Fedora

Guida alla Sicurezza

Guida alla protezione di Fedora Linux

Johnray Fuller

John Ha

David O'Brien

Scott Radvan

Eric Christensen

Adam Ligas

Fedora Guida alla Sicurezza
Guida alla protezione di Fedora Linux
Edizione 16.0

Autore	Johnray Fuller	jrfuller@redhat.com
Autore	John Ha	jha@redhat.com
Autore	David O'Brien	daobrien@redhat.com
Autore	Scott Radvan	sradvan@redhat.com
Autore	Eric Christensen	sparks@fedoraproject.org
Autore	Adam Ligas	adam@physco.com

La Guida alla Sicurezza intende assistere gli utenti Fedora ad apprendere i processi e le pratiche di messa in sicurezza di workstation e server da attività sospette, attacchi ed intrusioni, sia locali che remoti. La Guida, dedicata a sistemi Fedora, affronta concetti e tecniche valide su tutti i sistemi Linux, mostrando piani e gli strumenti necessari per creare un ambiente sicuro in postazioni domestiche, negli uffici e in centri di elaborazione dati. Con una gestione e un controllo adeguato, i sistemi Linux possono essere sia pienamente funzionali sia sicuri dai più comuni metodi di attacco e di intrusione.

Prefazione

1. Convenzioni del documento

Questo manuale utilizza numerose convenzioni per evidenziare parole e frasi, ponendo attenzione su informazioni specifiche.

Nelle edizioni PDF e cartacea questo manuale utilizza caratteri presenti nel set *Font Liberation*[1]. Il set Font Liberation viene anche utilizzato nelle edizioni HTML se il set stesso è stato installato sul vostro sistema. In caso contrario, verranno mostrati caratteri alternativi ma equivalenti. Da notare: Red Hat Enterprise Linux 5 e versioni più recenti, includono per default il set Font Liberation.

1.1. Convenzioni tipografiche

Vengono utilizzate quattro convenzioni tipografiche per richiamare l'attenzione su parole e frasi specifiche. Queste convenzioni, e le circostanze alle quali vengono applicate, sono le seguenti.

`Neretto monospazio`

Usato per evidenziare l'input del sistema, incluso i comandi della shell, i nomi dei file ed i percorsi. Utilizzato anche per evidenziare tasti e combinazione di tasti. Per esempio:

> Per visualizzare i contenuti del file **`my_next_bestselling_novel`** nella vostra directory di lavoro corrente, inserire il comando **`cat my_next_bestselling_novel`** al prompt della shell e premere **`Invio`** per eseguire il comando.

Quanto sopra riportato include il nome del file, un comando della shell ed un tasto, il tutto riportato in neretto monospazio e distinguibile grazie al contesto.

Le combinazioni di tasti possono essere distinte dai tasti tramite il trattino che collega ogni parte della combinazione. Per esempio:

> Premere **`Invio`** per eseguire il comando.
>
> Premere **`Ctrl`**+**`Alt`**+**`F2`** per smistarsi sul primo virtual terminal. Premere **`Ctrl`**+**`Alt`**+**`F1`** per ritornare alla sessione X-Windows.

Il primo paragrafo evidenzia il tasto specifico singolo da premere. Il secondo riporta due combinazioni di tasti, (ognuno dei quali è un set di tre tasti premuti contemporaneamente).

Se si discute del codice sorgente, i nomi della classe, i metodi, le funzioni i nomi della variabile ed i valori ritornati indicati all'interno di un paragrafo, essi verranno indicati come sopra, e cioè in **`neretto monospazio`**. Per esempio:

> Le classi relative ad un file includono **`filesystem`** per file system, **`file`** per file, e **`dir`** per directory. Ogni classe possiede il proprio set associato di permessi.

Proportional Bold

Ciò denota le parole e le frasi incontrate su di un sistema, incluso i nomi delle applicazioni; il testo delle caselle di dialogo; i pulsanti etichettati; le caselle e le etichette per pulsanti di selezione, titoli del menu e dei sottomenu. Per esempio:

[1] https://fedorahosted.org/liberation-fonts/

> Selezionare **Sistema**□□□ **Preferenze**□□□ **Mouse** dalla barra del menu principale per lanciare **Preferenze del Mouse**. Nella scheda **Pulsanti**, fate clic sulla casella di dialogo **mouse per mancini**, e successivamente fate clic su **Chiudi** per cambiare il pulsante primario del mouse da sinistra a destra (rendendo così il mouse idoneo per un utilizzo con la mano sinistra).
>
> Per inserire un carattere speciale in un file **gedit**, selezionare **Applicazioni**□□ **Accessori**□□□ **Mappa carattere** dalla barra menu principale. Successivamente, selezionare **Cerca**□□□ **Trova…** dalla barra del menu **Mappa carattere**, inserire il nome del carattere nel campo **Cerca** e cliccare **Successivo**. Il carattere ricercato verrà evidenziato nella **Tabella caratteri**. Fare un doppio clic sul carattere evidenziato per posizionarlo nel campo **Testo da copiare**, e successivamente fare clic sul pulsante **Copia**. Ritornare ora al documento e selezionare **Modifica**□□□ **Incolla** dalla barra del menu di **gedit**.

Il testo sopra riportato include i nomi delle applicazioni; nomi ed oggetti del menu per l'intero sistema; nomi del menu specifici alle applicazioni; e pulsanti e testo trovati all'interno di una interfaccia GUI, tutti presentati in neretto proporzionale e distinguibili dal contesto.

`Corsivo neretto monospazio` o ***`Corsivo neretto proporzionale`***

Sia se si tratta di neretto monospazio o neretto proporzionale, l'aggiunta del carattere corsivo indica un testo variabile o sostituibile . Il carattere corsivo denota un testo che non viene inserito letteralmente, o visualizzato che varia a seconda delle circostanze. Per esempio:

> Per collegarsi ad una macchina remota utilizzando ssh, digitare **`ssh`** ***`username@domain.name`*** al prompt della shell. Se la macchina remota è **`example.com`** ed il nome utente sulla macchina interessata è john, digitare **`ssh john@example.com`**.
>
> Il comando **`mount -o remount`** ***`file-system`*** rimonta il file system indicato. Per esempio, per rimontare il file system **`/home`**, il comando è **`mount -o remount /home`**.
>
> Per visualizzare la versione di un pacchetto attualmente installato, utilizzare il comando **`rpm -q`** ***`package`***. Esso ritornerà il seguente risultato: ***`package-version-release`***.

Da notare la parola in Corsivo neretto — nome utente, domain.name, file-system, pacchetto, versione e release. Ogni parola racchiude il testo da voi inserito durante l'emissione di un comando o per il testo mostrato dal sistema.

Oltre all'utilizzo normale per la presentazione di un titolo, il carattere Corsivo denota il primo utilizzo di un termine nuovo ed importante. Per esempio:

> Publican è un sistema di pubblicazione per *DocBook*.

1.2. Convenzioni del documento

Gli elenchi originati dal codice sorgente e l'output del terminale vengono evidenziati rispetto al testo circostante.

L'output inviato ad un terminale è impostato su **`tondo monospazio`** e così presentato:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Gli elenchi del codice sorgente sono impostati in **`tondo monospazio`** ma vengono presentati ed evidenziati nel modo seguente:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Note ed avvertimenti

E per finire, tre stili vengono usati per richiamare l'attenzione su informazioni che in caso contrario potrebbero essere ignorate.

Nota Bene

Una nota è un suggerimento o un approccio alternativo per il compito da svolgere. Non dovrebbe verificarsi alcuna conseguenza negativa se la nota viene ignorata, ma al tempo stesso potreste non usufruire di qualche trucco in grado di facilitarvi il compito.

Importante

Le caselle 'importante' riportano informazioni che potrebbero passare facilmente inosservate: modifiche alla configurazione applicabili solo alla sessione corrente, o servizi i quali necessitano di un riavvio prima di applicare un aggiornamento. Ignorare queste caselle non causa alcuna perdita di dati ma potrebbe causare irritazione e frustrazione da parte dell'utente.

Avvertenza

Un Avvertimento non dovrebbe essere ignorato. Se ignorato, potrebbe verificarsi una perdita di dati.

2. We Need Feedback!

More information about the Linux Security Guide project can be found at *https://fedorahosted.org/securityguide*

To provide feedback for the Security Guide, please file a bug in *https://bugzilla.redhat.com*[2]. Please select the proper component for this guide.

[2] https://bugzilla.redhat.com/enter_bug.cgi?component=security-guide&product=Fedora%20Documentation

Panoramica sulla Sicurezza

In seguito al sempre crescente affidamento di attività commerciali e di dati personali a sistemi di rete distribuiti, molte industrie del settore si sono organizzate fondando standard di sicurezza informatica. Le Aziende, per controllare la sicurezza dei loro sistemi e progettare soluzioni adatte alle loro esigenze operative, nel corso del tempo hanno sempre più richiesto la consulenza e le competenze di esperti di sicurezza. Molte aziende sono per natura dinamiche, con dipendenti che hanno accesso alle risorse IT della compagnia sia localmente sia da remoto, con la necessità di avere ambienti di elaborazione delle informazioni sicuri.

Sfortunatamente, molte organizzazioni (come pure i singoli utenti), considerano la sicurezza un aspetto secondario, un processo che viene tralasciato in favore di un aumento di efficenza, produttività e di entrate economiche. Spesso si pensa ad una vera pratica di sicurezza soltanto *dopo* che si è avuta un'intrusione. Gli esperti in sicurezza concordano che adottare alcune buone pratiche, prima di connettersi ad una rete poco sicura come Internet, è un mezzo efficace per contrastare molti tentativi di intrusione.

1.1. Introduzione alla Sicurezza

1.1.1. Cosa s'intende per Sicurezza Informatica?

Con Sicurezza Informatica si definisce un termine genarale che coinvolge un'ampia area dei processi informativi. Le aziende, per le loro transazioni economiche e per accedere ad informazioni strategiche, impiegano sistemi di computer e di rete, e considerano i dati trattati come una risorsa importante per la loro attività. Alcune definizioni e misurazioni di campo economico, come TCO (Total Cost of Ownership) o Costo Totale di Proprietà e QoS (Quality of Service) o Qualità del Servizio, rientrano anche nel nostro vocabolario. Attraverso questi strumenti, le aziende possono valutare integrità e disponibilità dei dati, come una parte dei costi nel processo di pianificazione e gestione. In alcune aziende, come nel commercio elettronico, la disponibilità e affidabilità dei dati può fare la differenza tra il succcesso e il fallimento aziendale.

1.1.1.1. Come è nata la Sicurezza Informatica?

La sicurezza dell'informazione si è evoluta nel corso degli anni, stimolata da una domanda di reti pubbliche in grado di mantenere riservate informazioni personali, finanziarie ed altri dati sensibili. Esistono numerose istanze come il caso Mitnick [1] e il caso Vladimir Levin [2], che hanno indotto molte organizzazioni industriali a ripensare ad un diverso modo di trattare l'informazione, la sua trasmissione e diffusione. La popolarità di Internet è stato uno degli sviluppi più importanti che ha portato a intensificare gli sforzi sulla sicurezza dei dati.

Un numero sempre crescente di persone usano i loro computer per accedere alle risorse offerte da Internet. Dalla ricerca e recupero di informazione alla posta elettronica, al commercio elettronico, Internet è stato riconosciuto come uno dei più importanti sviluppi del XX secolo.

Tuttavia, Internet e i suoi primi protocolli, sono stati sviluppati come un sistema *trust-based* o fidato. In altre parole, l'Internet Protocol non è stato progettato per essere sicuro. Non esistono nell'ambito degli stack di comunicazione TCP/IP degli standard di sicurezza approvati, risultando vulnerabile a potenziali utenti e processi maliziosi. Gli sviuppi moderni hanno reso la comunicazione su Internet più sicura, anche se di tanto in tanto, si verificano incidenti che conquistano l'attenzione mondiale e avvertono che nulla è ancora completamente sicuro.

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html
[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

1.1.1.2. La Sicurezza Oggi

Nel Febbraio del 2000, contro diversi siti Internet molto frequentati, fu portato un attacco di tipo DDoS (Distributed Denial of Service). L'attacco coinvolse yahoo.com, cnn.com, amazon.com, fbi.gov e diversi altri domini risultarono completamente isolati, irraggiungibili da parte dei normali utenti, poichè l'attacco riuscì a bloccare, per alcune ore, diversi router con raffiche di pacchetti ICMP molto lunghi, detti *ping flood*. L'attacco fu realizzato da un gruppo di anonimi che usarono dei programmi molto diffusi, appositamente sviluppati, per intercettare la presenza di porte vulnerabili nei server di rete; riuscirono ad installare sui server, delle applicazioni client, i *trojans*, e al momento giusto sferrarono un attacco contro ogni server infettato, rendendo i siti inutilizzabili. Da questa storia, molti concludono che la colpa sia nelle falle inerenti al sistema Internet, in quanto i router e i protocolli sono strutturati per accettare tutti i dati d'ingresso, a prescindere da dove vengano o del perchè siano stati spediti.

Nel 2007, una violazione di dati riuscì a compromettere la già nota debolezza del protocollo di cifratura per reti wireless, WEP (Wired Equivalent Privacy), causando la sottrazione, ai danni di una istituzione finanziaria mondiale, di oltre 45 milioni di numeri di carte di credito. [3]

In un altro caso, dall'auto del corriere, fu sottratto il disco che conteneva le registrazioni delle cedole assicurative di oltre 2,2 milioni di pazienti. [4]

Oggigiorno, circa 1,8 miliardi di persone nel mondo usano o hanno usato Internet. [5] Nello stesso tempo:

- Ogni giorno, secondo le registrazioni fornite dal CERT Coordination Center presso la Carnegie Mellon University.[6], si verificano circa 225 casi piuttosto gravi di falle di sicurezza.
- Nel 2003, il numero di casi riportati dal CERT è cresciuto a 137.529, dagli 82.094 nel 2002 e dai 52.658 nel 2001. [7]
- Il danno economico causato dall'impatto dei tre virus più pericolosi, diffusi su Internet negli ultimi tre anni, è di circa 13,2 miliardi di dollari.[8]

Da una indagine svolta nel 2008, per conto di *CIO Magazine* dal gruppo di esperti tecnologici e commerciali, "The Global State of Information Security"[9], sono emersi i seguenti punti:

- Appena il 43% degli intervistati analizzano o controllano la conformità degli utenti alle policy di sicurezza
- Soltanto il 22% mantiene un inventario delle aziende esterne che fanno uso dei loro dati
- Quasi la metà degli incidenti, dovuti a problemi di sicurezza, sono stati classificati come "Sconosciuti"
- Il 44% degli intervistati prevede di aumentare l'investimento in sicurezza nel prossimo anno
- Il 59% ritiene di avere una strategia di sicurezza informatica

Questi risultati sono una prova che la sicurezza informatica è diventata una spesa quantificabile e giustificabile negli investimenti IT. Le organizzazioni che richiedono integrità e pronta disponibilità

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/
[4] http://www.healthcareitnews.com/story.cms?id=9408
[5] http://www.internetworldstats.com/stats.htm
[6] http://www.cert.org
[7] http://www.cert.org/stats/fullstats.html
[8] http://www.newsfactor.com/perl/story/16407.html
[9] http://www.csoonline.com/article/454939/The_Global_State_of_Information_Security_

dei dati, sollecitano le competenze di amministratori di rete, sviluppatori ed ingegneri a garantire una affidabilità di 24h x 7giorni settimanali, ai loro sistemi, servizi ed informazioni. Cadere vittima di utenti o processi malintenzionati o di attacchi coordinati, è una minaccia al successo stesso dell'organizzazione.

Sfortunatamente, la sicurezza dei sistemi e della rete può risultare un affare piuttosto complicato, che richiede una conoscenza approfondita su come l'organizzazione considera, usa, manipola e trasmette le sue informazioni. Capire come un'organizzazione (e le persone che ne fanno parte) porta avanti i suoi affari è il punto di partenza per implementare un efficace progetto di sicurezza.

1.1.1.3. Standardizzare la Sicurezza

Le aziende di ogni settore si basano su regole e regolamenti che sono emanati da enti regolatori come l'IEEE (Institute of Electrical and Electronics Engineers). Lo stesso avviene per la sicurezza informatica. Molti consulenti e rivenditori del settore sicurezza informatica, concordano su un modello standard di sicurezza denominato CIA o *Confidentiality, Integrity and Availability*. Questo modello a tre livelli, è un componente generalmente accettato per stimare il rischio delle informazioni sensibili e per stabilire una policy di sicurezza. Di seguito si descrive il modello CIA in maggior dettaglio.

- Confidentiality — Le informazioni sensibili devono essere rese disponobili solo a un numero predefinito di persone. La trasmissione e l'uso non autorizzato di informazioni deve quindi essere limitato. Per esempio, la confidenzialità assicura che le informazioni finanziarie o personali di un cliente, non siano ottenute da un individuo non autorizzato, per propositi fraudolenti come la sostituzione d'identità o la sottrazione di credito.
- Integrity — L'informazione non deve essere alterata in modo da renderla incompleta o scorretta. Gli utenti non autorizzati non devono avere la possibilità di modificare o distruggere informazioni sensibili.
- Availability — L'informazione deve essere disponibile agli utenti autorizzati ogni qualvolta ciò è richiesto. La disponibilità è una garanzia che l'informazione può essere ottenuta sempre, in ogni momento. Questa è spesso misurata in termini percentuale e stabilita nei Service Level Agreement (SLA) in fase di contratto tra service provider e clienti.

1.1.2. SELinux

Fedora include un miglioramento al kernel Linux, denominato SELinux, che implementa una architettura MAC (Mandatory Access Control) per la regolazione precisa del controllo su file, processi, utenti ed applicazioni. Per ulteriori informazioni su SELinux, fare riferimento alla *Fedora SELinux User Guide*[10]. Per informazioni sulla configurazione e i servizi protetti da SELinux, consultare *Managing Confined Services*[11]. Per altre risorse, vedere il *Capitolo 8, Riferimenti*.

1.1.3. Controlli di Sicurezza

La Sicurezza Informatica è spesso suddivisa in tre categorie principali o *controls*:

- Fisico
- Tecnico
- Amministrativo

[10] http://docs.fedoraproject.org/it-IT/Fedora/13/html/Security-Enhanced_Linux/index.html
[11] http://docs.fedoraproject.org/en-US/Fedora/13/html/Managing_Confined_Services/index.html

Queste tre grandi categorie definiscono i principali obiettivi per una implemetazione di sicurezza. Nell'ambito di questi controlli, esistono delle sotto-categorie che ulteriormente suddividono i controlli e la loro implementazione.

1.1.3.1. Controlli Fisici

Il controllo fisico riguarda l'implementazione delle misure di sicurezza tali da impedire o prevenire accessi non autorizzati a materiale riservato. Esempi di controlli fisici includono:

- Video camere di sorveglianza a circuito chiuso
- Sistemi di allarme a sensore termico e di movimento
- Guardie di sicurezza
- Documenti d'identificazione
- Porte d'acciaio con serrature di sicurezza
- Sistemi Biometrici, tra cui strumenti di riconoscimento vocale e dell'iride, lettori di impronte digitali e facciali ed altri metodi usati per il riconoscimento degli individui

1.1.3.2. Controlli Tecnici

I controlli tecnici usano la tecnologia come base, per controllare l'accesso e l'uso di dati riservati in una struttura fisica e attraverso una rete. I controlli tecnici comprendono un'ampio ambito e diverse tecnologie, tra le quali:

- Tecniche di cifratura
- Smart card
- Autenticazione di rete
- Access control lists (ACLs)
- Software per controllare l'integrità dei file

1.1.3.3. Controlli Amministrativi

I controlli amministrativi definiscono i fattori umani legati alla sicurezza. Essi coinvolgono il personale di ogni livello di un'organizzazione e determinano quali utenti possono avere accesso a quali risorse ed informazioni, per mezzo di:

- Addestramento e consapevolezza
- Preparazione per affrontare disastri ed avviare piani di ripristino
- Strategie per assumere e licenziare il personale
- Registrazione e controllo di accesso del personale

1.1.4. Conclusione

Ora che si conoscono le origini, le ragioni e gli aspetti legati alla sicurezza, sarà più facile stabilire le azioni da intraprendere usando Fedora. Per poter pianificare ed implemetare una corretta strategia è importante individuare i fattori e le condizioni che garantiscono la sicurezza. Con queste informazioni,

il processo può essere formalizzato e la sua realizzazione diventa più chiara, man mano che si procede nei dettagli specifici del processo di sicurezza.

1.2. Attaccanti e Vulnerabilità

Per pianificare ed implementare una buona strategia di sicurezza, occorre conoscere i motivi che determinano, attaccanti motivati, ad avviare una intrusione nel sistema. Ma prima di affrontare questi motivi, bisogna introdurre la terminologia usata per identificare un attaccante.

1.2.1. Una breve storia degli Hacker

Il significato moderno della parola *hacker*, risale al 1960 ed al Tech Model Railroad Club del Massachusetts Institute of Technology (MIT), dove i membri si dilettavano a realizzare trenini elettrici, ricchi di dettagli e in diverse scale. *Hacker* era usato per indicare i membri del club che scoprivano un trucco o una ingegnosa scorciatoia per risolvere un problema.

Il termine hacker da allora è stato usato per descrivere sia gli appassionati di computer che i programmatori geniali. Una caratteristica che accomuna molti hacker è la curiosità di scoprire i dettagli di come funzionano i computer e le reti, senza una particolare motivazione ulteriore. Gli sviluppatori del software open source, spesso si considerano degli hacker, ed usano la parola hacker in senso di rispetto.

Solitamente, gli hacker seguono una forma di *etica hacker*, in cui è essenziale la ricerca e la conoscenza di informazione, e la condivisione di questa conoscenza con la community è uno dei doveri di ogni hacker. Con questa motivazione, spesso capita di sentire di sfide lanciate da hacker ai sistemi di sicurezza di computer di istituzioni universitarie. Per questo motivo, la stampa usa spesso il termine hacker, per indicare chiunque tenti di accedere illecitamente ai sistemi ed alla rete con intenzioni illecite, maliziose o criminali. In realtà la terminologia esatta per questo tipo di individuo è *cracker* — un termine appositamente creato dagli hacker, a metà degli anni '80, per ben differenziare le due comunità.

1.2.1.1. Tonalità di grigio

Negli Stati Uniti, si distinguono sostanzialmente tre tipi di gruppi che trovano e analizzano le vulnerabilità nei sistemi e nella rete. Questi gruppi sono spessso individuati dal colore del cappello che "indossano" quando eseguono un intervento, ed il colore è una indicazione del grado di rischio che stanno affrontando.

Chi porta un cappello di colore bianco o un *white hat hacker*, verifica le rete ed i sistemi valutando la loro performance e determinando quanto siano vulnerabili alle intrusioni. Di solito, un white hat hacker testa la sicurezza del sistema tentando di crackare il proprio sistema o quello di un cliente che lo ha appositamente chiamato. I ricercatori universitari e i consulenti in sicurezza, sono due esempi di white hat hacker.

Chi indossa un cappello di colore nero o un *black hat hacker*, è un cracker. In generale, i cracker non sono molto interessati alla programmazione o al funzionamento del sistema. Spesso si affidano a programmi maliziosi realizzati da altri, per carpire informazioni sensibili per scopi personali o causare danni ai sistemi ed alla rete.

Chi indossa un cappello grigio o un *gray hat hacker*, ha le competenze e, nella maggior parte dei casi, le intenzioni di un white hat hacker, ma occasionalmente utilizza le sue conoscenze con finalità meno nobili. Un gray hat hacker può essere immaginato come un white hat hacker che a volte, per propri motivi, diventa un black hat hacker.

Si può dire che un gray hat hacker segua un'altra etica hacker, secondo cui sarebbe lecito intrufolarsi nei sistemi, a patto di non commettere danni o carpire dati sensibili. Si potrebbe obbiettare,

comunque, che l'atto di intaccare un sistema è di per sè eticamente scorretto (n.d.t. oltre che legalmente perseguibile).

Qualunque sia l'intenzione di un intrusore, importante è conoscere le debolezze sfruttate dal cracker. Nella parte restante di questo capitolo ci si focalizzerà su questi aspetti.

1.2.2. Minacce alla sicurezza di rete

Pratiche scorrette quando si configurano i seguenti aspetti di rete, aumentano il rischio di un attacco.

1.2.2.1. Architetture non sicure

Una rete non correttamente configurata è il punto d'accesso principale per utenti non autorizzati. Una rete locale fidata ed *aperta* verso una rete altamente insicura come Internet, è vulnerabile come un'abitazione con una porta socchiusa in un quartiere a rischio — non è detto che succeda qualcosa, ma qualcuno potrebbe approfittare *eventualmente* della ingenuità.

1.2.2.1.1. Reti broadcast

Spesso gli amministratori di sistema trascurano, nei loro schemi di sicurezza, l'importanza dei dispositivi di rete. Semplici dispositivi come hub e router si basano sul principio di broadcast; cioè, quando un nodo trasmette un pacchetto ad un'altro nodo della rete, l'hub o il router invia in broadcast il pacchetto finchè il nodo destinatario non riceve e analizza il pacchetto. Questo metodo rende particolarmente vulnerabile *ARP* (Address Resolution Protocol) o *MAC* (Media Access Control) all'address spoofing da parte di intrusi sia esterni sia interni.

1.2.2.1.2. Server centralizzati

Un'altra potenziale trappola è l'uso di sistemi centralizzati. Un modo comunemente usato da molte aziende, per il contenimento dei costi, è quello di concentrare tutti i servizi su una singola macchina molto potente. Ciò può risultare conveniente, perchè facilita la gestione e riduce i costi di gestione, rispetto a configurazioni con server multipli. Tuttavia, un server centralizzato introduce un unico punto di rottura: se il server viene compromesso, ciò può portare all'inutilizzo completo della rete o peggio ancora, alla manomissione o sottrazione di dati. In queste situazioni, un server centrale diventa una porta aperta che permette di accedere all'intera rete.

1.2.3. Minacce alla sicurezza server

La sicurezza server è tanto importante quanto la sicurezza di rete, in quanto un server spesso gestisce moltissime informazioni vitali per un'organizzazione. Se un server viene compromesso, tutto il suo contenuto può diventare accessibile al cracker che può manometterlo o rubarlo. Le seguenti sezioni descrivono alcuni dei principali problemi.

1.2.3.1. Servizi non usati e porte aperte

Una installazione completa di Fedora comprende più di mille applicazioni e librerie. Comunque, molti amministratori di server non scelgono di installare tutti i pacchetti presenti nella distribuzione, preferendo invece una installazione di base con diverse applicazioni server.

Una pratica comune a molti amministratori, è installare il sistema operativo senza prestare attenzione a quali programmi vengono effetivamente installati. Ciò può causare futuri problemi, perchè si installano servizi non necessari, configurati con impostazioni predefinite ed eventualmente in esecuzione. Il risultato è di trovarsi con servizi non richiesti come Telnet, DHCP o DNS, in esecuzione su un server o workstation a insaputa dell'amministratore, che possono causare traffico indesiderato verso il server o peggio, una potenziale breccia nel sistema per i cracker. Fare riferimento alla

Sezione 2.2, «Server Security», per informazioni su come chiudere le porte e disabilitare i servizi non utilizzati.

1.2.3.2. Servizi privi di patch

Molte applicazioni server incluse in una installazione predefinita, risultano robuste ed ampiamente testate. Essendo state impiegate in ambienti di produzione per molti anni, il loro codice è stato estesamente rivisto e molti bug individuati e risolti.

Tuttavia, non esiste software perfetto e c'è sempre spazio per ulteriori rifiniture. Inoltre, il software più recente, spesso non sempre è rigorosamente testato come ci si aspetterebbe, vuoi perchè appena arrivato negli ambienti di produzione vuoi perchè non così comune come altre applicazioni server.

Gli amministratori di sistema insieme agli sviluppatori, spesso scoprono falle di vulnerabilità nelle applicazioni server e pubblicano le informazioni relative alla sicurezza, su mailing list come *Bugtraq*[12] o su siti come *Computer Emergency Response Team (CERT)*[13]. Sebbene questi meccanismi siano un metodo efficace per avvisare la comunità sui problemi di sicurezza, rimane comunque una responsabilità dell'amministratore provvedere a correggere reattivamente il proprio sistema. Ciò è particolarmente rilevante, in quanto anche i cracker hanno accesso ai suddetti servizi di informazione sulla sicurezza, ed useranno tali informazioni per attaccare i sistemi non corretti con ogni mezzo possibile. Quindi, in ottica di una maggiore sicurezza, a un amministratore di sistema si richiede vigilanza, tracciatura costante dei bug e appropriata manutenzione.

Per maggiori informazioni su come tenere aggiornato un sistema, vedere la *Sezione 1.5, «Aggiornamenti di sicurezza»*.

1.2.3.3. Amministrazione negligente

Gli amministratori che trascurano di correggere i loro sistemi, sono la prima grande minaccia per la sicurezza dei loro server. Secondo l'istituto *SANS* o SysAdmin, Audit, Network, Security Institute, la causa primaria che rende vulnerabile la sicurezza di un computer è *assegnare a personale impreparato la gestione della sicurezza e non fornire le risorse necessarie per l'addestramento*. [14] Ciò vale sia per gli amministratori senza esperienza sia per quelli troppo sicuri di sè o poco motivati.

Alcuni amministratori trascurano di applicare patch a server e workstation, altri di controllare i messaggi di log provenienti dal kernel o dal traffico di rete. Un altro errore comune si ha quando si lasciano invariate ai loro valori predefiniti, le password o le chiavi di acceso ai servizi. Per esempio, alcuni database hanno delle password di amministrazione predefinite, perchè si presume che l'amministratore cambi questa password immediatamente dopo l'installazione. Se un amministratore di database dimentica di cambiare questa password, anche un cracker inesperto usando una password predefinita a tutti nota, sarà in grado di guadagnare i privilegi di amministrazione sul database. Questi sono solo alcuni esempi di come una amministrazione poco attenta possa portare alla compromissione dei server.

1.2.3.4. Servizi intrinsecamente insicuri

Anche l'organizzazione più scrupolosa può diventare vittima di vulnerabilità, se i servizi di rete scelti sono intrinsecamente non sicuri. Per esempio, esistono molti servizi che sono sviluppati con l'assunzione che siano usati in reti fidate; quindi questa assunzione crolla nel momento in cui il servizio diventa disponibile su Internet — che è una rete intrinsecamente non fidata.

[12] http://www.securityfocus.com
[13] http://www.cert.org
[14] http://www.sans.org/resources/errors.php

Una categoria di servizi di rete insicuri sono quelli che richiedono l'autenticazione con username e password non cifrate. Telnet ed FTP sono due di tali servizi. Se uno sniffer di pacchetti si trova a monitorare il traffico, tra l'utente remoto e un tale servizio, esso può facilmente intercettare username e password.

Per loro natura, questi servizi possono molto facilmente cadere vittima di ciò che gli esperti di sicurezza definiscono con il termine, attacco *man-in-the-middle*. In questo tipo di attacco, un cracker una volta sabotato un name server, dirotta tutto il traffico sulla sua macchina. Quando l'utente apre una sessione remota con il server, la macchina dell'attaccante rimane trasparente, e silenziosamente situato *in mezzo* tra il servizio remoto e l'iconsapevole utente, può intercettare tutto il traffico. In questo modo, un cracker è in grado di carpire password e altri dati importanti, a insaputa del server e dell'utente.

Un'altra categoria di servizi insicuri includono NFS (Nework File Systems) e NIS (Network Information Services), sviluppati esplicitamente per l'impiego in LAN ma il cui uso, sfortunatamente, si è esteso alle WAN (per gli utenti remoti). NFS, per impostazione predefinita, non ha alcun meccanismo di autenticazione o sicurezza configurato per prevenire, da parte di un cracker, il montaggio del NFS e il conseguente accesso al suo contenuto. Anche NIS contiene informazioni, come password e permessi sui file, salvati in un file di testo ASCII in chiaro o (DBM ASCII-derived), che devono essere accessibili ad ogni computer della rete. Un cracker che riesce ad accedere al database può quindi scoprire ogni account utente sulla rete, incluso quello dell'amministratore.

Per impostazione predefinita, Fedora viene rilasciata con tutti questi servizi disattivati. Si tenga presente che nel caso occorra usare questi servizi, la loro accurata configurazione può risultare piuttosto critica. Per maggiori informazioni sulla configurazione ottimale dei servizi, fare riferimento alla *Sezione 2.2, «Server Security»*.

1.2.4. Minacce alla sicurezza di workstation e PC di casa

Workstation e PC non sono così frequentemente prede di attacchi come le reti o i server, ma siccome spesso contengono dati sensibili, come i dati relativi a carte di credito, essi possono diventare un obbiettivo dei cracker. Le workstation possono anche essere coinvolte ed usate, a insaputa dell'utente, come macchine "slave" per attacchi coordinati. Per queste ragioni, conoscere le vulnerabilità di workstation può evitare agli utenti la reinstallazione del sistema operativo o peggio, il difficile recupero dei dati trafugati.

1.2.4.1. Password inadeguate

Cattive password sono uno dei modi più semplici per agevolare ad un attaccante, l'accesso al sistema. Per saperne di più su come evitare di creare inutili falle con le password, fare riferimento alla *Sezione 2.1.3, «Protezione delle password»*.

1.2.4.2. Applicazioni client vulnerabili

Anche se un amministratore ha configurato e reso sicuro un server in maniera corretta, ciò non significa che un accesso remoto, da parte di un utente, sia sicuro. Per esempio, se il server permette l'accesso attraverso una rete pubblica, ai servizi Telnet od FTP, un attaccante potrebbe intercettare la username e la password trasmesse in chiaro, e quindi usare tali informazioni per accedere alla workstation dell'utente remoto.

Anche quando si usano protocolli sicuri come SSH, un utente remoto può essere vulnerabile a certi attacchi, se le applicazioni client non sono aggiornate. Per esempio, i client SSH della versione v.1, sono vulnerabili ad un attacco X-forwarding, da parte di server SSH maliziosi. Una volta connesso al server, l'attaccante può tranquillamente intercettare attraverso la rete, ogni tasto digitato od ogni click del mouse del client. Questo problema è stato risolto nella versione v.2 del protocollo SSH; in quasto

caso è un compito dell'utente sapere quali applicazioni soffrono di quali vulnerabilità ed aggiornarle, se necessario.

Nella *Sezione 2.1, «Workstation Security»*, si discute in maggior dettaglio i passi che amministratori ed utenti dovrebbero seguire, per limitare la vulnerabilità delle proprie workstation.

1.3. Analisi della vulnerabilità

Con a disposizione una buona dose di tempo, risorse e motivazione, un cracker può sabotare quasi ogni sistema. Alla fine di una giornata, tutte le procedure e tecnologie di sicurezza correntemente disponibili, non possono garantire che tutti i sistemi siano completamente salvi da intrusioni. I router aiutano a proteggere i gateway da Internet. I firewall aiutano a proteggere il confine della rete. I VPN (Virtual Private Networks) fanno passare i dati, in modo sicuro, in un flusso criptato. I sistemi anti-intrusione avvisano in caso di attività maliziose. Tuttavia, il successo di ciascuna di queste tecnologie dipende da un certo numero di variabili, tra cui:

- L'esperienza dello staff responsabile della configurazione, monitoraggio e mantenimento delle tecnologie.
- L'abiilità di coreggere ed aggiornare rapidamente ed efficacemente, servizi e kernel
- L'abilità dei responsabili di mantenere una vigilanza continua sulla rete.

Data la natura dinamica dei sistemi e delle tecnologie dell'informazione, rendere sicure le proprie risorse, può essere piuttosto complesso. A causa di questa complessità, risulta spesso difficile trovare degli esperti in tutti i settori del sistema. Se in un'azienda è possibile avere del personale con conoscenze generali in molte aree della sicurezza informatica, tuttavia, risulta difficile mantenere uno staff d'alto livello che sia esperto in ogni area. Questo perchè ciascuna area della sicurezza informatica richiede una attenzione costante e la sicurezza informatica risulta essere in continua evoluzione.

1.3.1. Pensare come il nemico

Si supponga di dover amministrare una rete aziendale. La rete generalmente comprende vari sistemi operativi, applicazioni, server, monitor di rete, firewall, sistemi anti-intrusione ed altro. Ora si immagini di provare a tenere aggiornati tutti questi sistemi. Vista la complessità dei software e delle reti attuali, gli attacchi e i bug sono una certezza. Mantenere al passo una intera rete con correzioni ed aggiornamenti, può essere una *impresa* in una grande organizzazione con sistemi etrogenei.

Si combini la richiesta di esperienza con il compito di essere al passo, ed inevitabilmente si verificheranno incidenti, i sistemi saranno compromessi, i dati corrotti ed i servizi interrotti.

Per migliorare le tecnologie relative alla sicurezza ed aiutare a proteggere i sistemi, le reti e i dati, occorre pensare come un cracker e valutare la sicurezza del proprio sistema, verificandone i punti di debolezza. Una valutazione preventiva della vulnerabilità del sistema e delle risorse di rete può rivelare potenziali problemi, che possono essere risolti prima che si verifichi un attacco.

Una valutazione della vulnerabilità è una verifica interna della sicurezza della rete e del sistema, i cui risultati indicano la confidenzialità, l'integrità e la disponibilità della rete (vedere la *Sezione 1.1.1.3, «Standardizzare la Sicurezza»*). Tipicamente, la valutazione inizia con una fase di ricognizione, durante la quale sono raccolti importanti dati riguardanti i sistemi e le risorse disponibili. Questa, porta alla fase di "readiness", in cui l'intero sistema è controllato in tutti i suoi punti di vulnerabilità. Essa culmina con la fase di reporting, in cui le vulnerabilità sono classificate in categorie di rischio alto, medio e basso; successivamente, si studiano i metodi per aumentare la sicurezza (o mitigare il rischio di vulnerabilità).

Se si facesse una valutazione di vulnerabilità della propria abitazione, si controllerebbero tutte le porte di casa per assicurarsi che siano chiuse e sicure. Si controllerebbero anche tutte le finestre, assicurandosi che siano chiuse e serrate. Lo stesso avviene con i sistemi, le reti e i dati informatici. Gli utenti maliziosi sono i ladri e i vandali dei dati. Occorre focalizzarsi sui loro strumenti, la loro mentalità e le loro motivazioni per poter reagire prontamente alle loro azioni.

1.3.2. Analisi e Test

L'analisi della vulnerabilità può essere svolta in due modalità: *Dall'esterno* e *Dall'interno*.

Quando si fa un'analisi di vulnerabilità dall'esterno, si tenta di compromettere il sistema dall'esterno. E' il punto di vista del cracker che non facendo parte della propria attività produttiva, si trova all'esterno. Si vede ciò che vede il cracker — indirizzi di routing pubblici, i sistemi presenti sulla *DMZ*, le interfacce esterne del firewall ed altro. DMZ sta per "zona demilitarizzata", corrispondente ad un computer o ad una piccola sottorete che si trova tra una rete interna fidata, come una LAN privata e una rete esterna non fidata, come Internet. Solitamente, una DMZ possiede dispositivi che accedono ad Internet, come server Web (HTTP), server FTP, server mail (SMTP) e server DNS.

Quando si fa un'analisi dall'interno, in un certo senso si è avvantaggiati, giacchè ci si trova all'interno e si gode della condizione di fiducia. Questo è il punto di vista che si acquista una volta loggati nel proprio sistema e che hanno anche i propri collaboratori all'interno della rete fidata. Si vedono server di stampa, file server, database ed altre risorse.

Tra le due modalità di analisi esistono nette differenze. All'interno della rete fidata si hanno maggiori privilegi di chiunque altro si trovi all'esterno. E ancora oggi, in molte organizzazioni, la sicurezza è vista come una intrusione dall'esterno, per cui viene configurata come se si volessse mantenere gli intrusori all'esterno. Molto poco viene fatto per proteggere le risorse interne (come firewall dipartimentali, controlli d'accesso sugli utenti, procedure d'autenticazione per accedere alle risorse interne ed altro). Solitamente, ci sono molte più risorse da analizzare in un'analisi interna piochè i principali sistemi si trovano all'interno. Una volta che si è fuori dall'organizzazione, si passa in uno stato non fidato. I sistemi e le risorse disponibili dall'esterno spesso sono molto limitate.

Si consideri la differenza tra analisi della vulnerabilità e *test di penetrazione* . Si pensi all'analisi di vulnerabilità come il primo passo per un test di penetrazione. L'informazione raccolta durante l'analisi viene usata per fare il test. Mentre l'analisi viene svolta per controllare la presenza di falle e potenziali vulnerabilità, il test di penetrazione praticamente ne verifica la loro pericolosità.

Analizzare le infrastrutture di rete è un processo dinamico. Anche la sicurezza dell'informazione e dei sistemi è un processo dinamico. Eseguendo un'analisi, si possono intercettare sia falsi positivi che falsi negativi.

Gli amministratori addetti alla sicurezza sono tanto validi quanto gli strumenti che usano e di cui sono a conoscenza. Si provi, per esempio, ad utlizzare uno degli strumenti di analisi disponibili, effettuando una verifica sul proprio sistema e quasi sicuramente si individueranno dei falsi positivi. Sia che si tratti di problemi nel programma o di un errore di utilizzo, l'effetto resta lo stesso. Lo strumento rileva vulnerabilità che in realtà non esistono (il falso positivo); o peggio ancora, non intercetta alcuna vulnerabilità che invece esiste (il falso negativo).

Quindi, ora che è stata definita la distinzione tra analisi della vulnerabilità e test di penetrazione, e la natura dei potenziali falsi negativi/positivi, in analisi future, prima di avviare un test di penetrazione, si rivedano attentamente i punti di vulnerabilità trovati.

Avvertimento

Tentare di sfruttare le vulnerabilità in un sistema di produzione può avere effetti negativi sulla produttività ed efficenza dell'intero sistema e della rete.

La seguente lista esamina alcuni benefici ricavabili da un'analisi di vulnerabilità:

- Crea un'attenzione proattiva verso la sicurezza informatica
- Individua potenziali falle prima dei cracker
- Consente di mantenere il sistema aggiornato e ben funzionante
- Promuove la crescita ed aiuta a sviluppare l'esperienza del team
- Abbatte le perdite economiche e la pubblicità negativa

1.3.2.1. Stabilre una metodologia

Per individuare gli strumenti da usare in un'analisi di vulnerabilità, può essere utile stabilire una metodologia di analisi della vulnerabilità. Sfortunatamente, al momento non esiste una metodologia predefinita o standardizzata; ad ogni modo, il buon senso e una buona pratica possono essere una guida sufficiente.

Qual'è l'obbiettivo? Si sta controllando un solo server o l'intera rete con tutti i suoi sistemi? Siamo all'interno o all'esterno della nostra organizzazione? Le risposte a queste domande sono importanti perchè aiutano a stabilire non solo quali strumenti usare ma anche come usarli.

Per saperne di più su come stabilire una metodologia, fare riferimento ai seguenti siti:

- *The Open Source Security Testing Methodology Manual* (OSSTMM): *http://www.isecom.org/osstmm* [15]
- *The Open Web Application Security Project*: *http://www.owasp.org/* [16]

1.3.3. Valutazione degli strumenti

Un'analisi inizia dalle informazioni raccolte da un qualche strumento. Quando si analizza una intera rete conviene dapprima crearsi una mappa, per sapere gli host che sono in esecuzione. Una volta localizzati, si esamini ogni host, individualmente. La loro analisi richiederà, probabilmente, altri strumenti. Sapere quali strumenti usare può essere il passo più cruciale in un'analisi di vulnerabilità.

Proprio come nella vita di tutti i giorni, esistono molti strumenti differenti che svolgono lo stesso lavoro. La stessa situazione si ha quando si affronta un'analisi di vulnerabilità. Esistono strumenti specifici per i sistemi operativi, le applicazioni ed anche per le reti (a seconda del protocollo usato). Alcuni sono free, altri no. Alcuni strumenti sono intuitivi e facili da usare, altri sono critpici e scarsamente documentati ma con proprietà che altri non hanno.

[15] http://www.isecom.org/osstmm/
[16] http://www.owasp.org/

Trovare gli strumenti giusti può essere piuttosto scoraggiante all'inizio e un po' d'esperienza può contare molto. Se possibile, impostare un sistema di test e si provino più strumenti possibile, notando i punti di forza e debolezza di ciascuno. Di ogni strumento si legga il README o le pagine man relative. Si cerchi anche su Internet articoli, guide passo-passo, o mailing-list dedicate allo strumento.

Gli strumenti elencati sono solo un piccolo campione di quelli disponibili.

1.3.3.1. Scansione degli Host con Nmap

Nmap è uno strumento incluso in Fedora che può essere usato per determinare il layout di una rete. Nmap è disponibile da molti anni ed è probabilmente lo strumento più usato per raccogliere informazioni. Una notevole pagina man provvede a fornire una dettagliata descrizione sul suo uso e le sue opzioni. Gli amministratori possono usare Nmap su una rete per individuare gli host presenti ed aprire le porte di questi sistemi.

Nmap è uno strumento molto adatto per un'analisi di vulnerabilità. Esso è in grado di creare una mappa di tutti gli host all'interno della rete e, passando un opzione, è possibile conoscere anche il sistema operativo in esecuzione su un particolare host. Nmap è un buon punto di partenza per creare una policy che usi servizi sicuri e blocchi quelli non utilizzati.

1.3.3.1.1. Usare Nmap

Nmap può essere avviato da un terminale con il comando **nmap**, seguito dall'hostname o dall' indirizzo IP della macchina di cui si vuole eseguire una scansione.

```
nmap foo.example.com
```

I risultati di una scansione base (che potrebbe durare anche un paio di minuti, dipendendo da dove sia localizzato l'host e da altre condizioni di rete), dovrebbero essere qualcosa di simile:

```
Starting Nmap 4.68 ( http://nmap.org )
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT STATE SERVICE
22/tcp open ssh
53/tcp open domain
70/tcp closed gopher
80/tcp open http
113/tcp closed auth
```

Nmap testa le più comuni porte di comunicazione in attesa o ascolto di servizi. Questa informazione può aiutare un amministratore a chiudere servizi non necessari o inutilizzati.

Per maggiori informazioni sull'uso di Nmap, fare riferimento alla homepage ufficiale, al seguente URL:

http://www.insecure.org/ [17]

1.3.3.2. Nessus

Nessus è uno scanner di sicurezza. L'architettura a plug-in di Nessus permette di personalizzare il suo utilizzo, secondo le necessità della rete e del sistema. Come ogni scanner, Nessus rimane

[17] http://www.insecure.org/

uno strumento valido finchè rimane valido il database delle firme. Fortunatamente, Nessus è frequentemente aggiornato ed offre report completi, scansione degli host e ricerca in tempo reale di vulnerabilità. Si ricordi che potrebbero rivelarsi falsi positivi e falsi negativi, anche in uno strumento potente e frequentemente aggiornato come Nessus.

Nota

Il client e il server Nessus è disponibile nei repository di Fedora ma il suo uso richiede una iscrizione. Nessus è stato inserito in questo documento come riferimento per quegli utenti che potrebbero essere interessati ad usare questa diffusa applicazione.

Per maggiori informazioni su Nessus, fare riferimento al sito web ufficiale, al seguente URL:

http://www.nessus.org/ [18]

1.3.3.3. Nikto

Nikto è uno scanner di scrpit CGI (Common Gateway Interface). Nikto controlla le vulnerabilità in script CGI, ma in modo da essere evasivo così da eludere i sistemi anti-intrusione. Prima di usarlo, si consiglia di leggere attentamente la documentazione allegata alla sua distribuzione. Se si dispone di un server Web che serve script CGI, Nikto può essere una eccellente risorsa per controllare la sicurezza di questi server.

Maggiori informazioni su Nikto, possono trovarsi al seguente URL:

http://www.cirt.net/code/nikto.shtml [19]

1.3.3.4. VLAD lo scanner

VLAD è uno scanner di vulnerabilità sviluppato dal gruppo RAZOR presso Bindview, Inc., che controlla la Top Ten dei problemi di sicurezza più comuni (probelmi SNMP, di condivisione file, ecc), nella lista SANS. Anche se non così ricco di funzionalità come Nessus, VLAD è comunque un buon investigatore.

Nota

VLAD non è incluso in Fedora e non è supportato. E' stato inserito in questo documento come riferimento per quegli utenti che potrebbero essere interessati ad usare questa diffusa applicazione.

Maggiori informazioni su VLAD, possono trovarsi sul sito web di RAZOR, al seguente URL:

http://www.bindview.com/Support/Razor/Utilities/ [20]

[18] http://www.nessus.org/
[19] http://www.cirt.net/code/nikto.shtml
[20] http://www.bindview.com/Support/Razor/Utilities/

1.3.3.5. Le necessità future

Per ogni target e risorsa esistono molti strumenti disponibili. Esistono strumenti per reti wireless, reti Novell, sistemi windows, sistemi Linux ed altri ancora. Un altro aspetto importante da considerare, quando si analizzano le vulnerabilità, riguarda la sicurezza fisica, la selezione del personale e l'analisi delle reti vocali/PBX. Nuovi concetti come *war walking*, riguardanti la scansione perimetrale della struttura fisica in cui ha sede l'organizzazione, alla ricerca di vulnerabilità nelle reti wireless, sono alcuni concetti emergenti che si potrebbero investigare, e se necessario, includere in un'analisi di routine. L'immaginazione, il tempo e le risorse sono gli unici limiti per pianificare e condurre un'analisi di vulnerabilità.

1.4. Rischi e Attacchi comuni

La *Tabella 1.1, «Attacchi comuni»* illustra alcune delle azioni più comuni e i punti d'ingresso usati per accedere alle risorse di rete di un'organizzazione. Per ogni attacco si fornisce una descrizione di come sia stata realizzata e le contromisure da prendere, a protezione delle risorse di rete.

Tabella 1.1. Attacchi comuni

Attacco	Descrizione	Note
Password vuote o predefinite	Lasciare le password amministrative vuote oppure utilizzare una password predefinita, impostata dal produttore. Ciò è molto comune in alcuni hardware come router e firewall ed anche in alcuni servizi in esecuzione su Linux (in Fedora invece non esistono password predefinite).	Si trovano comunemente in hardware di rete come router, firewall, VPN e dispositivi di memorizzazione di rete (NAS). Comune in molti sistemi operativi proprietari, specialmente in quelli che vendono servizi (come UNIX e Windows). Gli amministratori a volte creano account di utenti privilegiati, in fretta e furia, lasciando la password vuota; ciò può essere un punto d'accesso ideale per utenti maliziosi che scoprono l'account.
Chiavi predefinite condivise	Alcuni servizi di sicurezza, a volte, per motivi di sviluppo o per test valutativi, impostano le chiavi di sicurezza in modo predefinito. Se le chiavi non vengono modificate e vengono usate in un ambiente di produzione su Internet, *tutti* gli utenti con le stesse chiavi predefinite avranno accesso alle risorse di quella chiave ed alle informazioni sensibili che essa contiene.	Molto comune negli access point dei sistemi wireless e nelle appliance secure server preconfigurate.
IP Spoofing	Una macchina remota agisce come un nodo sulla rete locale, trova le vulnerabilità nei server ed installa un programma backdoor o trojan, per ottenere il controllo sulle risorse di rete.	Lo spoofing è abbastanza difficile da realizzare, dato che comporta prevedere, da parte dell'attaccante, i numeri della sequenza TCP/IP necessari per coordinare una connessione con il sistema target; tuttavia, sono disponibili molti strumenti che assistono i cracker nel perseguire questo tipo di attacco.

Attacco	Descrizione	Note
		Dipende dai servizi in esecuzione sul sistema target (come **`rsh`**, **`telnet`**, FTP e altri) che usano tecniche di autenticazione *source-based*, i quali non sono raccomandati se confrontati con PKI o altre forme di autenticazione cifrata, usate in **`ssh`** o SSL/TLS.
Eavesdropping (Origliare)	Raccogliere dati che passano tra i nodi attivi di una rete, stando in ascolto fra i due nodi della connessione.	Questo tipo di attacco funziona, principalmente, nei protocolli con trasmissione del testo in chiaro come Telnet, FTP ed HTTP. Gli attaccanti remoti, per eseguire questo attacco, devono avere accesso ad un sistema compromesso sulla LAN; solitamente, il cracker usa un attacco attivo (come l'IP spoofing o man-in-the-middle), per compromettere il sistema sulla LAN. Misure preventive includono servizi con scambio di chiavi crittografiche, password "usa e getta" oppure autenticazione cifrata; è inoltre consigliata una robusta cifratura durante la trasmissione.
Vulnerabilità nei servizi	L'attaccante può trovare una falla o una scappatoia in un servizio in esecuzione su Internet; attraverso questa vulnerabilità, l'attaccante compromette l'intero sistema e qualsiasi dato in esso contenuto, e potrebbe compromettere altri sistemi sulla rete.	I servizi basati su HTTP come CGI, sono vulnerabili all'esecuzione di comandi remoti ed anche ad accessi da shell interattive. Anche se il servizio HTTP è in esecuzione come un utente non privilegiato, come "nobody", informazioni come file di configurazione e mappe di rete possono essere lette, oppure l'attaccante può avviare un attacco tipo DoS (Denial of Service) consumando risorse di sistema o renderle indisponibili agli utenti. A volte i servizi possono presentare vulnerabilità che non vengono trovate in fase di sviluppo e di test; queste vulnerabilità (come i *buffer overflows*, in cui l'attaccante manda in crash un servizio, riempiendo il buffer di memoria di una applicazione, con valori arbitrari e all'attaccante danno un prompt di comando interattivo, dal quale può eseguire comandi arbitrari), possono fornire un controllo amministrativo completo ad un attaccante.

Attacco	Descrizione	Note
		Gli amministratori dovrebbero assicurarsi che i servizi non siano in esecuzione come utente root, e dovrebbero vigilare su patch e aggiornamenti di errata per le applicazioni, da produttori o da organizzazioni di sicurezza come il CERT e il CVE.
Vulnerabilità nelle applicazioni	L'attaccante trova falle nelle applicazioni desktop e workstation (come i client e-mail) per eseguire codice arbitrario, impiantare *trojan* per attacchi futuri o per mandare in crash il sistema. Potrebbero verificarsi ulteriori attacchi, se la workstation compromessa ha privilegi amministrativi sul resto della rete.	Le workstation e i desktop sono più facili da sfruttare se gli utenti non hanno le conoscenze o l'esperienza per prevenire o rilevare un rischio; è importante informare gli utenti sui rischi che si corrono, quando si installa software non autorizzato oppure si aprono allegati di mail non attese. Si possono implementare dei metodi di sicurezza, facendo in modo che i software di gestione posta non aprano o eseguano automaticamente gli allegati. In aggiunta, l'aggiornamento automatico delle workstation tramite i servizi di rete Red Hat o altri servizi di gestione, possono ridurre il carico di lavoro e le disattenzioni sulla sicurezza in sistemi multi-utente.
Attacchi Denial of Service (DoS)	Gli attaccanti o gruppi di attaccanti si coordinano contro la rete di una organizzazione o contro le risorse di un server, inviando pacchetti non autorizzati all'host obiettivo (può essere un server, un router o una workstation). Ciò induce la risorsa a diventare non disponibile agli utenti legittimi.	Il caso più famoso di DoS si è verificato negli USA nel 2000. Molti siti commerciali e di governo ad alto traffico, sono stati resi in-disponibili da un attacco coordinato di ping flood usando diversi sistemi compromessi a banda larga, che agivano da *zombie* o nodi rimbalzanti di pacchetti broadcast. Il mittente dei pacchetti, di solito, viene falsificato (oltre ad essere ritrasmesso) rendendo arduo scoprire l'origine dell'attacco. Migliorare il filtraggio dei pacchetti in ingresso (IETF rfc2267), usando **`iptables`** e sistemi di intrusione (IDS) come **`snort`**, possono aiutare gli amministratori a individuare e prevenire attacchi DoS distribuiti.

1.5. Aggiornamenti di sicurezza

Se viene scoperto una vulnerabilità di sicurezza, il software colpito deve essere aggiornato per ridurre qualsiasi rischio connesso. Se il software fa parte di un pacchetto di Fedora, correntemente supportato, Fedora si impegna a rilasciare, prima possibile, gli aggiornamenti di correzione. Spesso, gli avvisi su un problema di sicurezza si accompagnano con una patch (una porzione di codice che

risolve il problema). Questa patch, una volta applicata al pacchetto e testata, viene poi rilasciata come aggiornamento di correzione. Altre volte, quando un avviso non include una patch, lo sviluppatore lavora insieme con il manutentore del software per risolvere il problema. Poi una volta risolto, il pacchetto viene testato e rilasciato come aggiornamento di correzione.

Se viene rilasciato un aggiornamento di correzione per il software in uso, si raccomanda di applicare l'aggiornamento prima possibile, in modo da ridurre la potenziale vulnerabilità del sistema.

1.5.1. Aggiornare i pacchetti

Quando si aggiorna un sistema, è importante scaricare gli aggiornamenti da una sorgente fidata. Un attaccante può facilmente ricompilare un pacchetto con lo stesso numero di versione di quello che si suppone risolva il problema, ma con un'azione differente sulla sicurezza, per poi rilasciarlo su Internet. Anche usando misure di sicurezza, come la verifica dell'integrità dei file, non ci si accorgerebbe della minaccia presente nel pacchetto contraffatto. Quindi, è molto importante scaricare gli RPM soltanto da sorgenti fidate, come Fedora, e controllare la firma del pacchetto per verificarne l'integrità.

Nota

Fedora include una conveniente icona nel pannello del desktop, che si allerta quando è disponibile un aggiornamento per il sistema Fedora.

1.5.2. Verificare la firma dei pachetti

Tutti i pacchetti di Fedora sono firmati con la chiave *GPG* di Fedora. GPG sta per GNU Privacy Guard o GnuPG, ossia un software libero usato per assicurare l'autenticità dei file distribuiti. Per esempio, una chiave privata (segreta) sigilla il pacchetto mentre la chiave pubblica apre e verifica il pacchetto. Se la chiave pubblica, distribuita da Fedora, non corrisponde con la chiave privata durante la verifica di RPM, il pacchetto potrebbe essere stato alterato e perciò non è attendibile.

L'utility RPM, presente in Fedora, prova a verificare la firma GPG di un pacchetto RPM, prima di procedere alla sua installazione. Se la firma GPG di Fedora non è stata installata, installarla da un repository sicuro, per esempio da un DVD di installazione di Fedora.

Supponendo che il disco sia montato su **`/mnt/cdrom`**, usare il seguente comando per importare la firma nel *keyring* (un database di chiavi fidate presenti nel sistema):

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Per visualizzare l'elenco di tutte le chiavi installate, per la verifica RPM, eseguire il comando:

```
rpm -qa gpg-pubkey*
```

L'output sarà qualcosa di simile:

```
gpg-pubkey-db42a60e-37ea5438
```

Per visualizzare i dettagli di una chiave, usare il comando **`rpm -qi`** seguito dall'output del comando precedente, come indicato di seguito:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

E' molto importante verificare la firma dei file RPM, prima di procedere all'installazione, per essere sicuri che non sinao stati alterati. Per verificare tutti i pacchetti scaricati, eseguire il seguente comando:

```
rpm -K /tmp/updates/*.rpm
```

Per ciascun pacchetto, se la chiave GPG viene verificata con successo, il comando restituisce **`gpg OK`**. Diversamente, assicurarsi di usare la chiave pubblica di Fedora e verificare la sorgente da cui sono stati scaricati i pacchetti. I pacchetti che non superano la verifica GPG non dovrebbero essere installati, poichè potrebbero essere stati alterati da terze parti.

Dopo aver verificato la chiave GPG e scaricato tutti i pacchetti di correzione, procedere con l'installazione come utente root.

1.5.3. Installare pacchetti firmati

L'installazione di molti pacchetti (esclusi quelli del kernel), si esegue con il seguente comando

```
rpm -Uvh /tmp/updates/*.rpm
```

Per i pacchetti del kernel usare il seguente comando:

```
rpm -ivh /tmp/updates/<kernel-package>sshd.
```

Sostituire<*kernel-package*> con il pacchetto RPM del kernel.

Una volta riavviata la macchina, usare il nuovo kernel; il vecchio kernel può essere rimosso, con il seguente comando:

```
rpm -e <old-kernel-package>
```

Sostituire <*old-kernel-package*> con il pacchetto RPM del kernel da rimuovere.

Nota

Non è strettamente necessario rimuovere il vecchio kernel. Il gestore di boot, GRUB, permette di avere kernel multipli, selezionabili da un menu nella fase di boot.

Importante

Prima di installare una correzione di sicurezza, leggere le istruzioni nell'avviso di correzione allegato alla patch e poi procedere come indicato. Per istruzioni generali su come applicare le modifiche, in un aggiornamento di correzione, fare riferimento alla *Sezione 1.5.4, «Applicare i cambiamenti»*.

1.5.4. Applicare i cambiamenti

Dopo aver scaricato ed installato gli aggiornamenti di correzione e di sicurezza, è importante chiudere e riavviare qualsiasi software oggetto di aggiornamento. Ciò ovviamente dipende dal tipo di software

aggiornato. La seguente lista mostra le varie categorie di software e indica come usare la versione aggiornata.

Nota

In generale, il riavvio del sistema resta il modo più sicuro che garantisce che si stia usando la versione appena aggiornata; comunque il riavvio non sempre è richiesto o disponibile all'amministratore.

Applicazioni

Le applicazioni dello spazio utente sono tutti quei programmi avviabili da un utente. Solitamente, tali applicazioni sono usate soltanto quando un utente, uno script o una utilty automatizzata le avvia e non persistono per lunghi periodi di tempo.

Una volta aggiornata un'applicazione, chiudere ogni istanza dell'applicazione presente nel sistema e riavviare l'applicazione in modo da usare la versione aggiornata.

Kernel

Il kernel è il nucleo centrale del sistema operativo Fedora. Esso gestisce l'accesso alla memoria, il processore, le periferiche e organizza tra loro i vari componenti citati.

Data la sua centralità, il kernel non può essere riavviato senza riavviare la macchina. Perciò, una versione aggiornata del kernel non può essere usata se non si riavvia la macchina.

Librerie condivise

Le librerie condivise sono pezzi di codice, come **`glibc`**, usate da applicazioni e servizi. Le applicazioni che utilizzano una libreria condivisa, di solito caricano il codice condiviso durante l'inizializzazione dell'applicazione, perciò le applicazioni che usano una libreria che è stata aggiornata devono essere chiuse e riavviate.

Per determinare quali applicazioni sono collegate ad una libreria, usare il comando **`lsof`** come indicato:

```
lsof /lib/libwrap.so*
```

Il comando restituisce un elenco di tutti i programmi in esecuzione che usano involucri (wrapper) TCP per il controllo d'accesso. Perciò, tutti i programmi in elenco devono essere fermati e riavviati nel caso in cui il pacchetto **`tcp_wrappers`** venga aggiornato.

Servizi SysV

I servizi SysV sono programmi server persistenti, avviati durante il processo di boot. Esempi di Servizi SysV includono **`sshd`**, **`vsftpd`**, e **`xinetd`**.

Poichè questi servizi, generalmente persistono in memoria dopo il boot, ogni servizio SysV aggiornato deve essere fermato e riavviato. Ciò può essere fatto usando **Sistema > Amministrazione > Servizi**, oppure eseguendo il comando **`/sbin/service`**, da una shell di root, come indicato di seguito:

```
/sbin/service <service-name> restart
```

Nel precedente esempio, sostituire *`<service-name>`* con il nome del servizio, per esempio **`sshd`**.

Servizi **`xinetd`**

I servizi controllati dal super servizio **`xinetd`** sono in esecuzione soltanto se è attiva una connessione. Esempi di servizi controllati da **`xinetd`** includono Telnet, IMAP e POP3.

Poichè nuove istanze di questi servizi sono avviati da **`xinetd`** ogni volta che viene ricevuta una nuova richiesta, le connessioni che si attivano dopo un aggiornamento sono gestite dal software aggiornato. Invece, le connessioni attive precedenti all'aggiornamento continuano ad essere gestite dalla versione precedente.

Per arrestare (kill) le vecchie istanze di un servizio controllato da **`xinetd`**, aggiornare il pacchetto del servizio e poi arrestare tutti i processi in esecuzione. Per sapere se il processo è in esecuzione usare il comando **`ps`** e poi il comando **`kill`** o **`killall`**, per arrestare tutte le istanze correnti del servizio

Per esempio, se viene rilasciato un aggiornamento di sicurezza per il pacchetto **`imap`**, aggiornare il pacchetto e poi eseguire il seguente comando in una shell di root:

```
ps -aux | grep imap
```

Questo comando restituisce tutte le sessioni IMAP attive. Le sessioni individuali possono essere chiuse con il seguente comando:

```
kill <PID>
```

Se con il precedente comando la sessione non si chiude, usare allora il seguente comando:

```
kill -9 <PID>
```

Nei precedenti esempi, sostituire *`<PID>`* con l'ID del processo (l'ID del processo si trova nella seconda colonna del comando **`ps`**), della sessione IMAP.

Per chiudere tutte le sessione IMAP attive, eseguire il comando:

```
killall imapd
```

Proteggere la rete locale

2.1. Workstation Security

La sicurezza di un ambiente Linux inizia dalle workstation. La policy di sicurezza deve partire dalla singola macchina, in modo da assicurare la sicurezza alla macchina e al sistema di cui la macchina fa parte. Un rete di computer è sicura soltanto se non esiste alcun punto debole.

2.1.1. Analizzare la sicurezza di una workstation

Quando si analizza la sicurezza di una workstation Fedora, occorre tener conto dei seguenti fattori:

- *Sicurezza del BIOS e del Boot Loader* — Può un utente non autorizzato accedere fisicamente alla macchina ed avviare la macchina in modalità mono utente o di ripristino, senza usare una password?
- *Sicurezza della Password* — Quanto sono sicure le password di accesso degli utenti?
- *Controlli Amministrativi* — Chi può accedere al sistema e quanti controlli amministrativi possiede?
- *Servizi di rete disponibili* — Quali servizi sono in ascolto per servire richieste dalla rete: devono essere tutti in esecuzione?
- *Firewall* — Che tipo di firewall, se occorre, è necessario?
- *Strumenti di comunicazione sicuri* — Quali strumenti dovrebbero essere usati per le comunicazioni tra workstation e quali evitati?

2.1.2. Protezione del BIOS e del Boot Loader

Proteggere con password BIOS e Boot Loader, impedisce ad utenti non autorizzati di avviare la macchina con dischi di avvio o di ottenere privilegi amministrativi, in modalità single user. Le misure da prendere servono sia a proteggere le informazioni nella macchina sia la macchina stessa.

For example, if a machine is used in a secure location where only trusted people have access and the computer contains no sensitive information, then it may not be critical to prevent such attacks. However, if an employee's laptop with private, unencrypted SSH keys for the corporate network is left unattended at a trade show, it could lead to a major security breach with ramifications for the entire company.

2.1.2.1. Password per accedere al BIOS

Le ragioni per proteggere il BIOS di un compter con password, sono fondamentalmente due, [1]:

1. *Impedire le modifiche alle impostazioni del BIOS* — Se un intrusore ha accesso al BIOS, egli può configurare l'avvio da USB o DVD, permettendogli di avviare la modalità rescue del sistema o la modalità single user, con possibilità di avviare processi arbitrari o copiare dati sensibili.
2. *Impedire il Boot di sistema* — Alcuni BIOS permettono di proteggere con password, il processo di boot. Se attivato, all'accensione della macchina viene richiesto di inserire una passowrd. In tal modo, un attacker deve conoscere la password per avviare il processo di boot.

[1] Il numero e il tipo di protezione supportata dipende dai produttori

I metodi per l'impostazione della password di BIOS variano tra produttori, consultare perciò il manuale della motherboard allegato al computer, per informazioni specifiche.

La password di BIOS può essere resettata, disconnettendo la pila CMOS o agendo sui ponticelli di contatto nella motherboard: per questo motivo, si consiglia di rendere inaccessibile, per quanto possibile, il case del computer. Comunque, prima di manovrare sulla motherboard, fare riferimento ai manuali a disposizione.

2.1.2.1.1. Rendere sicure le piattaforme non-x86

Altre architetture usano degli assembler con operazioni hardware di basso livello, grosso modo simili al BIOS dei sistemi x86. Per esempio, le macchine con processori Intel® Itanium™ usano la shell *Extensible Firmware Interface* (*EFI*).

Per istruzioni su come proteggere con password, i simil-BIOS di altre architetture, fare riferimento alle indicazioni del produttore.

2.1.2.2. Password per Boot Loader

Le ragioni principali per proteggere con password, un boot loader Linux sono le seguenti:

1. *Impedire l'accesso Single User Mode* — Se un attacker può avviare il sistema in modalità mono utente, egli accede automaticamente come utente root senza che venga richiesta la passoword di root.

2. *Impedire l'accesso alla console GRUB* — Se la macchina usa GRUB come proprio boot loader, un attacker può usare l'interfaccia di editazione di GRUB per modificare la configurazione o per carpire informazioni, con il comando **`cat`**.

3. *Impedire l'accesso a sistemi operativi poco sicuri* — In un sistema dual boot, un attacker può selezionare un sistema operativo privo di policy di controllo d'accesso e di permessi, come DOS.

Nelle piattaforme x86, Fedora viene distribuito con il boot loader GRUB. Per informazioni dettagliate su GRUB, fare riferimento alla **Fedora Installation Guide** su *http://docs.fedoraproject.org*.

2.1.2.2.1. Proteggere GRUB con password

Per configurare GRUB secondo le richieste della *Sezione 2.1.2.2, «Password per Boot Loader»*, aggiungere una direttiva di password al suo file di configurazione. Le operazioni da eseguire sono, scegliere per prima cosa una password robusta, aprire un terminale, avviando una shell di root, e poi digitare il seguente comando:

```
/sbin/grub-md5-crypt
```

Quando richiesto, inserire la password per GRUB e premere **`Invio`**. Il comando restituisce un hash MD5 della password.

Successivamente, aprire il file di configurazione di GRUB, **`/boot/grub/grub.conf`** e inserire, immediatamente dopo la riga contente la stringa **`timeout`** nella sezione principale del file, la seguente riga:

```
password --md5 <password-hash>
```

[2] GRUB accetta anche password in chiaro, tuttavia per aumentare il livello di sicurezza si raccomanda di aggiungere un hash MD5

Sostituire <*password-hash*> con il valore restituito dal comando **`/sbin/grub-md5-crypt`**[2].

Al successivo riavvio del sistema, il menu di GRUB vieta l'accesso all'interfaccia di editazione o di comando, se non dopo aver digitato **`p`** seguito dalla password di GRUB.

Per impostare la terza richiesta, ossia impedire in un sistema dual boot l'avvio di un s.o. poco sicuro, occorre editare sempre il file **`/boot/grub/grub.conf`**.

Nella riga contenente la stringa **`title`**, individuare il sistema operativo che si vuole proteggere ed aggiungere immediatamente dopo, la direttiva **`lock`**.

Per un sistema DOS, la riga diventerebbe qualcosa di simile:

```
title DOS lock
```

Attenzione

Perchè questo metodo funzioni correttamente, occorre che sia presente una riga **`password`**, nella sezione principale del file **`/boot/grub/grub.conf`**. Diversamente, un attacker potrebbe accedere all'interfaccia di editazione di GRUB e rimuovere il lock.

Per creare una password diversa per ogni kernel o sistema operativo, aggiungere **`lock`**, seguito dalla password, su ogni riga relativa.

Ogni sistema protetto da una password dovrebbe iniziare con una riga simile:

```
title DOS lock password --md5 <password-hash>
```

2.1.3. Protezione delle password

Le password sono il metodo principale usato da Fedora per verificare l'dentità di un utente. Per questo motivo, la sicurezza della password è molto importante: serve a proteggere l'utente, la workstation e la rete.

Per motivi di sicurezza, il processo di installazione configura il sistema usando *Message-Digest Algorithm* (*MD5*) e password non leggibili. Si raccomanda vivamente di non alterare queste impostazioni.

Se durante l'installazione, si deseleziona la codifica MD5, le password saranno generate usando il vecchio formato *Data Encryption Standard* (*DES*). Questo standard, limita le password ad otto caratteri alfanumerici (vietando l'uso di caratteri di punteggiatura e di altri caratteri speciali), con un modesto livello di codifica a 56 bit.

Inoltre se si deseleziona l'illeggibilità delle password, le password saranno salvate e cifrate con un funzione hash one-way, nel file **`/etc/passwd`** accessibile a tutti, rendendo il sistema vulnerabile ad attacchi da parte di cracker di password. Infatti, se un intrusore riesce ad accedere ad una macchina come un regolare utente, egli può copiare il file **`/etc/passwd`** sulla propria macchina, e carpire le password salvate, sebbene cifrate, usando una delle tante applicazioni di cracking disponibili. A questo punto è solo una questione di tempo: se è presente una password poco sicura, l'applicazione prima o poi riuscirà facilmente a decodificarla.

Le password illeggibili eliminano questo tipo di attacco, salvando le password cifrate nel file **`/etc/shadow`**, leggibile soltanto da parte dell'utente root.

Un potenziale attacker può tentare di carpire le password anche da remoto, tramite un servizio di rete attivo sulla macchina come SSH o FTP. Questo tipo di attacco richiede più tempo e lascia traccia nei file di log del sistema. Ma in presenza di *password deboli*, a suo favore, il cracker che inizia un attacco contro un sistema, p.e in piena notte, potrebbe avere accesso al sistema prima dell'alba, e tempo sufficiente per cancellare nel file di log, ogni traccia dei suoi tentativi d'accesso.

Oltre al formato e al salvataggio che sono considerazioni di sistema, c'è il problema del contenuto, che è la cosa effettivamente fondamentale che spetta all'utente, ossia creare una password robusta.

2.1.3.1. Creare password robuste

Per creare una password sicura è una buona idea seguire queste linee guida:

- *Non usare solo parole o solo numeri* — In una password usare una miscela di parole e numeri (Sull'uso delle parole vedi più avanti).

  Ecco alcuni esempi di password poco sicure:

  - 8675309
  - antonio
  - hackme

- *Non usare parole riconoscibili* — Parole come nomi propri, sostantivi o anche termini di show televisi o di attori, anche se terminanti con dei numeri, dovrebbero essere evitati.

  Ecco alcuni esempi di password poco sicure:

  - bisio45
  - jolie-34
  - mazingaZ

- *Non usare parole di lingue straniere* — Le applicazioni di cracking, spesso, scansionano le parole nei dizionari di molte lingue straniere. Affidarsi a una parola straniera non è molto sicuro.

  Ecco alcuni esempi di password poco sicure:

  - cheguevara
  - bienvenido1
  - 1dumbKopf

- *Non usare la terminologia Hacker* — Se si ritiene di rientrare in una elite, perchè per la propria password usa la terminologia Hacker — anche chiamato linguaggio l337 (LEET) — si rifletta bene. Molti dizionari includono il linguaggio 1337.

  Ecco alcuni esempi di password poco sicure:

  - H4X0R
  - 1337

- *Non usare informazioni personali* — Evitare di usare ogni informazione personale. Se l'attacker conosce un pò l'identità della vittima, il suo compito di deduzione della password si semplifica. La seguente lista mostra il genere di password da evitare:

Ecco alcuni esempi di password poco sicure:

  - Il proprio nome
  - I nomi dei propri animali domestici
  - I nomi dei familiari
  - Le date di nascita
  - Il proprio numero di telefono o codice postale

- *Non invertire parole riconoscibili* — Buoni programmi di cracking sono capaci di invertire parole comuni, per cui invertire una password debole non ne aumenta la sicurezza.

  Ecco alcuni esempi di password poco sicure:

  - R0X4H
  - oinotna
  - 43-eiloj

- *Non trascrivere la password* — Mai conservare una password su un pezzo di carta. Meglio impararla a memoria!
- *Non usare la stessa password su tutte le macchine* — Su ogni macchina usare una password differente. In questo modo, se un sistema viene compromesso, le altre macchine non sono immediatamente a rischio.

Di seguito si riportano alcuni suggerimenti per creare password robuste:

- *Creare password lunghe almeno otto caratteri* — Più lunga la password, tanto meglio. Se si usa la codifica MD5, la password dovrebbe essere lunga almeno 15 caratteri. Con la codifica DES usare la lunghezza massima (otto caratteri).
- *Usare lettere maiuscole e minuscole* — Fedora è case sensitive (distingue tra maiuscole/ minuscole), per cui l'uso di lettere miste aumenta la robustezza delle password.
- *Usare lettere e numeri* — L'aggiunta di numeri alle password, soprattutto se inserite all'interno (non solo all'inizio o alla fine), aumenta la robustezza delle password.
- *Includere caratteri speciali* — L'uso di caratteri speciali, come &, $, e >, può notevolmente migliorare la robustezza di una password (ciò non è possibile con la codifica DES).
- *Scegliere una password da ricordare* — La miglior password del mondo serve a ben poco, se poi non si può ricordare; usare acronimi o altre tecniche di memorizzazione, per tenere a mente la password.

Con tutte queste regole, può sembrare difficile creare una password che soddisfi tutti i criteri di una buona password, evitando tutte le caratteristiche di una cattiva. Fortunatamente, esistono alcuni procedimenti per creare una password, sicura e facile da ricordare.

2.1.3.1.1. Metodologia per creare password sicure

Esistono diversi metodi per creare password sicure. Uno dei più comuni impiega acronimi. Ecco un esempio:

- Si pensi ad una frase facile da ricordare, come

  con un mazzo di rose rosse, fischiettando, vado all'appuntamento con la mia bella

- Successivamente, trasformare la frase, inclusa la punteggiatura, in un acronimo.

  cumdrr,f,vaaclmb

- Aggiungere un pò di "rumore" sostituendo, numeri e simboli al posto delle lettere. Per esempio, sostituire, la **a** con **7** e la **d** con il simbolo at (@):

  cum@rr,f,v77clmb

- Aggiungere ulteriore "rumore", capitalizzando almeno una lettera, per esempio la **m**.

  cum@rr,f,v77clMb

- Non usare mai come password, la *riproduzione fedele* di questo esempio.

Se è imperativo creare password sicure, la loro corretta gestione è altrattanto importante, soprattutto per gli amministratori di organizzazioni più grandi. Il paragrafo seguente, illustrerà buone pratiche per creare e gestire le password degli utenti di una organizzazione.

2.1.3.2. Creare le password degli utenti di una organizzazione

Se un'organizzazione ha un gran numero di utenti, gli amministratori di sistema hanno a disposizione due opzioni di base per incoraggiare l'uso di buone password. Possono creare le password per i loro utenti oppure possono lasciare agli utenti la creazione delle proprie password, verificando che esse siano qualitativamente accettabili.

La creazione delle password da assegnare agli utenti, assicura che esse siano buone ma alla lunga può appesantire, soprattutto se l'organizzazione manifesta una certa dinamicità nel turn over del personale. Inoltre ciò aumenta il rischio che gli utenti appuntino la password su carta.

Per questi motivi, la maggior parte degli amministratori peferisce lasciare agli utenti la creazione delle proprie password, per poi verificare attivamente che siano buone ed in alcuni casi, obbligare gli utenti a cambiarle periodicamente, usando delle password con validità temporale limitata.

2.1.3.2.1. Obbligare ad usare password robuste

Per proteggere la rete da intrusioni, è buona norma per gli amministratori verificare che le password usate all'interno dell'organizzazione siano robuste. Quando gli utenti devono creare o modificare la password, essi possono usare l'applicazione **passwd** gestito da *Pluggable Authentication Manager* (*PAM*), in grado di verificare se la password digitata è troppo corta o facile da crackare. Questa verifica avviene tramite il modulo PAM, **pam_cracklib.so**. Poichè PAM è configurabile, è possibile aggiungere altri moduli di verifica delle password, come **pam_passwdqc** (disponibile su *openwall.com* [3]) o anche realizzare un nuovo modulo. Per una lista dei moduli PAM disponibili, fare riferimento a *PAM modules*[4] sul sito di kernel.org. Per maggiori informazioni su PAM, fare riferimento alla *Sezione 2.4, «Pluggable Authentication Modules (PAM)»*.

La verifica fatta all'atto di creazione della password, tuttavia, non rileva password cattive così efficacemente come invece fanno le applicazioni di cracking.

[3] http://www.openwall.com/passwdqc/

[4] http://www.kernel.org/pub/linux/libs/pam/modules.html

Sono disponibili molte applicazioni di cracking che funzionano su Fedora, anche se nessuna viene distribuita con il sistame operativo. Di seguito viene fornito un elenco delle più comuni applicazioni di cracking:

- ***John The Ripper*** — Un'applicazione di cracking, flessibile e veloce. Permette di usare più liste di parole e, tramite ricerca esaustiva (o forza bruta) di crackare le password. L'applicazione è disponibile sul sito *openwall.com*[5].
- ***Crack*** — Forse l'applicativo di cracking più conosciuto, **Crack** è anche molto veloce, sebbene non così semplice da usare come **John The Ripper**. Può essere trovato sul sito *crypticide.com*[6].
- ***Slurpie*** — **Slurpie**, simile a **John The Ripper** ed a **Crack**, è stato progettato per essere eseguito contemporaneamente su più computer, in modo da creare un sistema di cracking distribuito. Può essere trovato, insieme ad altri strumenti di attacco che operano su sistemi distribuiti, su *ussrback.com*[7].

Attenzione

Assicurarsi sempre di avere le necessarie autorizzazioni, prima di tentare qualsiasi cracking di password, nella propria organizzazione.

2.1.3.2.2. Passphrase

Nei sistemi moderni, le passphrase (o frasi d'accesso) e le password, sono le pietre angolari della sicurezza. Sfortunatamente, tecniche ben più sicure ed affidabili come biometrie o autenticazioni a due fattori, ancora non fano parte di molti sistemi. Se le password vengono impiegate per rendere sicuro un sistema, occorre spiegare il ruolo svolto dalle passphrase. Queste ultime sono più lunghe delle password e permettono una migliore protezione rispetto alle password, anche quando vengono implementate senza usare caratteri non-standard, come numeri e simboli.

2.1.3.2.3. Durata delle password

Limitare la durata delle password, è un'altra tecnica usata dagli amministratori di sistema per proteggere l'organizzazione da cattive password. Con tale tecnica, dopo un determinato periodo di tempo (generalmente 90 giorni), all'utente viene richiesto di ricreare una nuova password. La teoria che giustifica tutto ciò è che, se un utente è obbligato a cambiare periodicamente la propria password, allora una password crackata rimane utile ad un intrusore, soltanto per un periodo di tempo limitato. L'aspetto negativo è che potrebbe aumentare la tendenza dell'utente a trascrivere su carta, la propria password.

In Fedora sono disponibili due applicazioni usate per impostare la durata di una password: il comando **`chage`** e l'applicazione grafica **Gestione Utenti** (**`system-config-users`**).

L'opzione **`-M`** nel comando **`chage`**, permette di specificare il numero di giorni di validità della password. Per esempio, per impostare la scadenza di una password dopo 90 giorni, usare il seguente comando:

```
chage -M 90 <username>
```

[5] http://www.openwall.com/john/
[6] http://www.crypticide.com/alecm/security/crack/c50-faq.html
[7] http://www.ussrback.com/distributed.htm

Nel comando precedente, sostituire <*username*> con il nome dell'utente. Per disabilitare la scadenza su una password, è consuetudine usare il valore **99999** (equivalente a circa 273 anni).

Per modificare scadenze e informazioni di più account, si può usare il comando **chage** in modo interattivo. Per entrare in modalità interattiva, digitare il segente comando:

```
chage <username>
```

Di seguito si riporta un esempio di sessione interattiva:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@myServer ~]#
```

Per maggiori informazioni sulle opzioni disponibili, fare riferimento alle pagine di man.

Per impostare scadenze su password, si può usare anche l'applicazione grafica **Gestione Utenti**. Nota: occorre essere amministratore per effettuare questa operazione.

1. Per avviare l'interfaccia Gestione Utenti, selezionare dal menu **Sistema > Amministrazione > Utenti e Gruppi**. Oppure in un terminale, digitare il comando **system-config-users**.
2. Selezionare la scheda, **Utenti** e quindi l'utente interessato, nella lista degli utenti.
3. Per visualizzare la finestra delle Proprietà dell'Utente, cliccare sul bottone **Proprietà**, (oppure dal menu, selezionare **File > Proprietà**).
4. Selezionare la scheda **Password Info** e abilitare la casella di contollo con l'etichetta, **Abilitare la scadenza sulla password**.
5. Inserire il valore richiesto nel campo **Giorni di validità** e poi cliccare sul bottone **OK**.
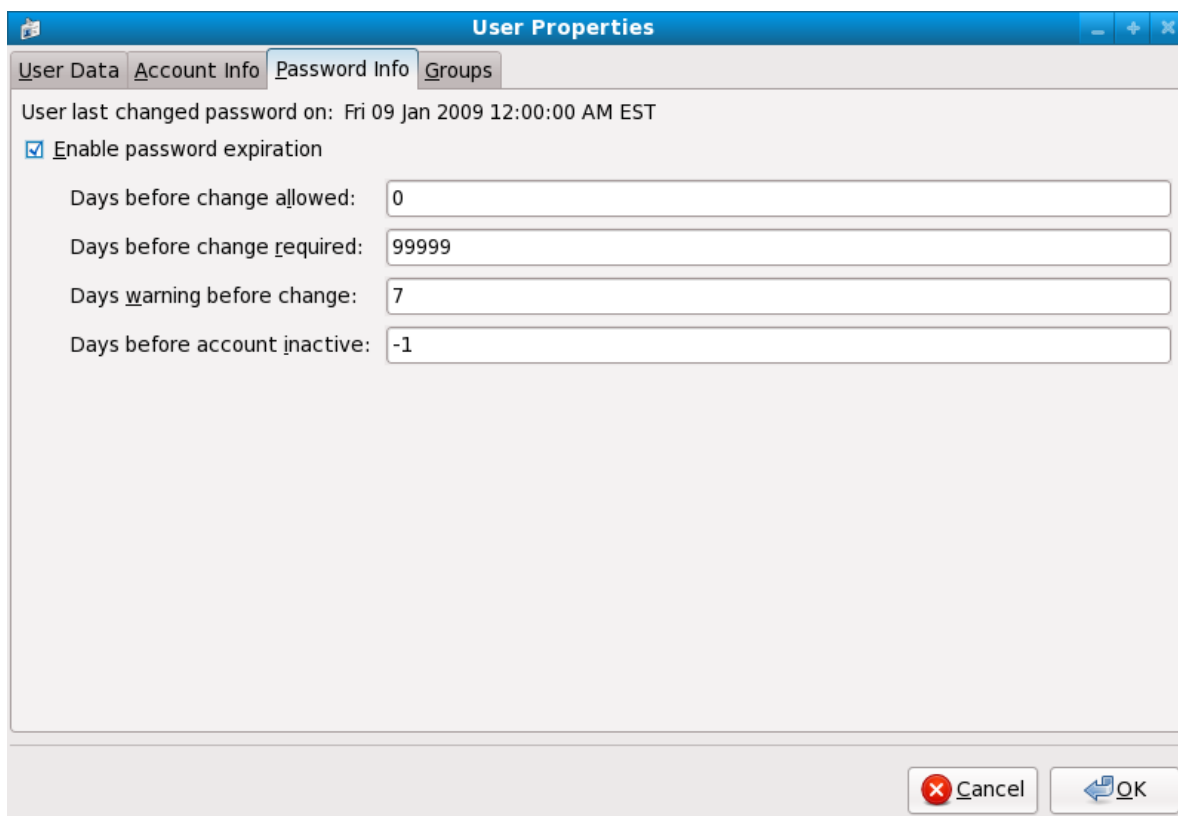
Figura 2.1. Impostazione della scadenza

2.1.4. Controlli amministrativi

Quando si gestisce un PC, per esempio il PC di casa, l'utente può svolgere i compiti di amministrazione come utente root, oppure acquisire privilegi effettivi di root, con programmi *setuid*, come **sudo** o **su**. Un programma setuid opera con l'ID utente (o *UID*) del proprietario del programma, e non con l'UID di colui che utilizza il programma. Questi programmi, in un listato di formato lungo, sono denotati con una **s** nei flag di proprietà, come indicato di seguito:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```

Nota

La **s** può essere maiuscola o minuscola. Se è maiuscola vuol dire che il bit di permesso non è stato impostato.

Nell'ambito di una organizzazione, gli amministratori devono stabilire se e quali tipi di accessi amministrativi assegnare agli utenti delle proprie macchine. Per esempio, attraverso il modulo PAM denominato **pam_console.so**, alcuni compiti normalmente riservati soltanto all'utente root, come il riavvio o il montaggio di supporti rimovibili, sono estesi al primo utente che accede ad un terminale (fare riferimento alla *Sezione 2.4, «Pluggable Authentication Modules (PAM)»*, per maggiori informazioni sul modulo **pam_console.so**). Inoltre, altri importatnti compiti amministrativi, come modificare le impostazioni di rete, configurare un nuovo mouse o montare un dispositivo di rete, sono possbili soltanto se si hanno i privilegi necessari. Quindi, gli amministratori di sistemi, devono stabilire il livello di accesso da attribuire agli utenti della rete aziendale.

2.1.4.1. Permettere l'accesso come utente root

Se gli utenti di una organizzazione sono fidati ed adeguatamente esperti, allora il loro accesso come root non dovrebbe essere un problema. Permettere di accedere come root, significa assegnare agli utenti attività di minore importanza, come aggiungere dispositivi o configurare interfacce di rete, lasciando agli amministratori maggiore libertà per aspetti più importanti, come garantire la sicurezza della rete e del sistema.

Dall'altro lato, permettere ai singoli utenti l'accesso come utente root, può generare i seguenti problemi:

- *Errata configurazione della macchina* — Gli utenti con accesso privilegiato, potrebbero configurare erroneamente la propria macchina e richiedere la necessaria assistenza. Peggio ancora, potrebbero causare, inconsapevolmente, delle falle nella sicurezza del sistema.
- *Eseguire servizi non sicuri* — Gli utenti con accesso root, potrebbero eseguire sulle proprie macchine, servizi insicuri come FTP o Telnet, mettendo potenzialmente a rischio le loro credenziali di accesso, ossia username e password. Infatti, questi servizi trasmettono in chiaro queste informazioni nella rete.
- *Inviare allegati e-mail come root* — Sebbene piuttosto rari, si può dire che non esistono virus allegati in email, che possano minacciare un sistema Linux. L'unica situazione che può rivelarsi una minaccia, si ha quando gli allegati vengono aperti dall'utente root.

2.1.4.2. Disabilitare l'accesso come utente root

Se per queste o altre ragioni, un amministratore ritiene opportuno non dover assegnare agli utenti i privilegi di root, allora la password di root dovrebbe essere custodita segretamente, e l'accesso al runlevel 1 o l'accesso *single user mode*, dovrebbe essere disabilitato (vedere la *Sezione 2.1.2.2, «Password per Boot Loader»*, per maggiori ragguagli su questo tipo di protezione).

La *Tabella 2.1, «Metodi per disabilitare l'account root»* descrive altri metodi disponibili all'amministratore, per disabilitare gli accessi come utente root:

Tabella 2.1. Metodi per disabilitare l'account root

Metodo	Descrizione	Influenza	Non influenza
Modificare la shell di root	Aprire il file **`/etc/passwd`** e modificare la shell da **`/bin/bash`** in **`/sbin/nologin`**.	Vieta l'accesso alla shell di root e registra nei file log di sistema, ogni tentativo d'accesso. I seguenti programmi *non possono accedere* all'account root: · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · **`su`** · **`ssh`** · **`scp`** · **`sftp`**	Programmi che non necessitano di una shell, come client FTP, e-mail e molti programmi setuid. I seguenti programmi *possono accedere* all'account root: · **`sudo`** · client FTP · client e-mail
Disabilitare l'accesso root da ogni	Un file **`/etc/securetty`** vuoto, nega l'accesso come utente root, da	Vieta l'accesso all'account root da un terminale locale o da remoto. I seguenti	I programmi che non eseguono come root, ma eseguono compiti

Metodo	Descrizione	Influenza	Non influenza
terminale (tty)	qualsiasi terminale collegato al computer.	programmi *non possono accedere* all'account root: · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · Altri servizi di rete che aprono un tty	amministrativi attraverso setuid o altri meccanismi. I seguenti programmi *possono accedere* all'account root: · **`su`** · **`sudo`** · **`ssh`** · **`scp`** · **`sftp`**
Disabilitare gli accessi SSH di root	Aprire il file **`/etc/ssh/sshd_config`** e impostare il parametro **`PermitRootLogin`** su **`no`**.	Vieta l'accesso all'account root via gli strumenti OpenSSH. I seguenti programmi *non possono accedere* all'account root: · **`ssh`** · **`scp`** · **`sftp`**	Il metodo vieta l'accesso all'account root, soltanto attraverso gli strumenti OpenSSH.
Usare PAM per limitare l'accesso all'account root da parte dei servizi.	Nella directory **`/etc/pam.d/`**, modificare il file relativo al servizio interessato. Assicurarsi che per l'autenticazione sia richiesto il file **`pam_listfile.so`**.[1]	Vieta l'accesso all'account root ai servizi di rete controllati da PAM. I seguenti servizi *non possono accedere* all'account root: · client FTP · client e-mail · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · **`ssh`** · **`scp`** · **`sftp`** · Tutti i servizi controllati da PAM	I programmi e i servizi non controllati da PAM.

[1] Fare riferimento alla *Sezione 2.1.4.2.4, «Disabilitare l'account root usando PAM»* per i dettagli.

2.1.4.2.1. Disabilitare la shell di root

Per evitare che gli utenti accedano direttamente come root, l'amministratore di sistema può impostare nel file **`/etc/passwd`**, la shell dell'account root su **`/sbin/nologin`**. Ciò impedisce di accedere all'account root, con i comandi che richiedono una shell, come **`su`** e **`ssh`**.

Importante

I programmi che non necessitano di accedere alla shell, come client e-mail o il comando **`sudo`**, tuttavia possono continuare ad accedere all'account root.

2.1.4.2.2. Disabilitare le sessioni di root

Per ulteriormente limitare l'accesso all'accout root, gli amministratori possono disabilitare le sessioni di root da terminale, modificando il file **`/etc/securetty`**. Questo file elenca tutti i dispositivi da cui l'utente root può avviare una sessione. Se il file non esiste, allora l'utente root può avviare una sessione da ogni tipo di dispositivo di comunicazione presente, sia via terminale sia attraverso una interfaccia di rete. Ciò potrebbe essere piuttosto rischioso per la sicurezza della rete, giacchè si potrebbe avviare una sessione come utente root, via Telnet, servizio che trasmette in chiaro le informazioni di accesso. In Fedora, per impostazione, il file **`/etc/securetty`** permette di avviare una sessione di root, soltanto attraverso un terminale fisicamente collegato alla macchina. Per vietare ogni tipo di sessione di root, rimuovere il contenuto di questo file, digitando il seguente comando:

```
echo > /etc/securetty
```

Attenzione

Un file **`/etc/securetty`** completamente vuoto, *consente* tuttavia di avviare sessioni di root da remoto, usando l'insieme di strumenti OpenSSH, poichè il terminale non viene aperto fino ad autenticazione avvenuta.

2.1.4.2.3. Disabilitare le sessioni SSH di root

Le sessioni di root, attraverso il protocollo SSH, in Fedora sono disabilitate per impostazione; comunque, se questa impostazione viene abilitata può essere nuovamente disabilitata, modificando il file di configurazione del demone SSH (**`/etc/ssh/sshd_config`**). Modificare la riga:

```
PermitRootLogin yes
```

con la seguente:

```
PermitRootLogin no
```

Per rendere effettive le modifiche, riavviare il demone SSH, per esempio con il seguente comando:

```
kill -HUP `cat /var/run/sshd.pid`
```

2.1.4.2.4. Disabilitare l'account root usando PAM

PAM, con il modulo **`/lib/security/pam_listfile.so`**, permette di regolare in maniera flessibilie gli accessi degli account. L'amministratore può usare questo modulo, per creare una lista di utenti non autorizzati ad avviare sessioni. Il file di configurazione **`/etc/pam.d/vsftpd`**, nel seguente esempio, mostra un utilizzo del modulo sul server FTP, **`vsftpd`** (il carattere **`\`** alla fine della prima riga, *non* è necessario se la direttiva rientra in un'unica riga):

```
auth required /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Con questa istruzione, PAM legge il file **`/etc/vsftpd.ftpusers`** in cui sono elencati tutti gli utenti a cui è vietato l'accesso al servizio. L'amministratore può modificare il nome di questo file, mantenere una lista separata per ogni servizio oppure usare una lista unica per vietare l'accesso a più servizi.

Se un amministratore vuole negare l'accesso a più servizi, un'analoga riga può essere aggiunta ai file PAM di configurazione, come **/etc/pam.d/pop** e **/etc/pam.d/imap** per client e-mail o **/etc/pam.d/ssh** per client SSH.

Per maggiori informazioni su PAM, fare riferimento alla *Sezione 2.4, «Pluggable Authentication Modules (PAM)»*.

2.1.4.3. Limitare l'accesso all'account root

Piuttosto che negare completamente l'accesso all'utente root, l'amministratore potrebbe limitare l'accesso solo ai programmi setuid, come **su** o **sudo**.

2.1.4.3.1. Il comando su

Quando si esegue il comando **su**, viene richiesto di inserire la password di root, e dopo autenticazione si ha a disposizione una shell di root.

Una volta avviata la sessione con il comando **su**, l'utente *è* l'utente root, con pieno ed assoluto controllo sul sistema.[8] Inoltre, una volta diventato root, l'utente può usare il comando **su** per diventare altri utenti presenti nel sistema, senza che sia richiesta alcuna password.

Data la grande potenza di questo programma, gli amministratori potrebbero limitarne l'accesso ad un numero ristretto di utenti.

Uno dei modi più semplici per far ciò, consiste nell'aggiungere gli utenti scelti, ad un gruppo amministrativo speciale, denominato *wheel*. In concreto, come utente root digitare il seguente comando:

```
usermod -G wheel <username>
```

Nel precedente comando, sostituire <*username*> con lo username dell'utente che si vuole aggiungere al gruppo **wheel**.

Alternativamente, si può usare la GUI **Gestione Utenti** per modificare il gruppo di appartenenza degli utenti, come spiegato di seguito. Nota: Occorre possedere i privilegi di amministratore per effettuare questa operazione.

1. Per avviare l'interfaccia Gestione Utenti, selezionare dal menu **Sistema > Amministrazione > Utenti e Gruppi**. Oppure in un terminale, digitare il comando **system-config-users**.

2. Selezionare la scheda, **Utenti** e quindi l'utente interessato, nella lista degli utenti.

3. Per visualizzare la finestra delle Proprietà dell'Utente, cliccare sul bottone **Proprietà**, (oppure dal menu, selezionare **File > Proprietà**).

4. Selezionare la scheda **Gruppi**, nella lista attivare la checkbox relativa al gruppo wheel e poi cliccare sul bottone **OK**. Fare riferimento alla *Figura 2.2, «Aggiungere utenti al gruppo "wheel"»*.

5. In un editor di testo, aprire il file di configurazione PAM per il comando **su** (**/etc/pam.d/su**) e rimuovere il carattere di commento **#**, dalla seguente riga:

   ```
   auth  required /lib/security/$ISA/pam_wheel.so use_uid
   ```

[8] Questo accesso è ancora soggetto alle restrizioni imposte da SELinux, se abilitato

Questa modifica comporta che soltanto i membri del gruppo di amministrazione **`wheel`** possono usare questo programma.
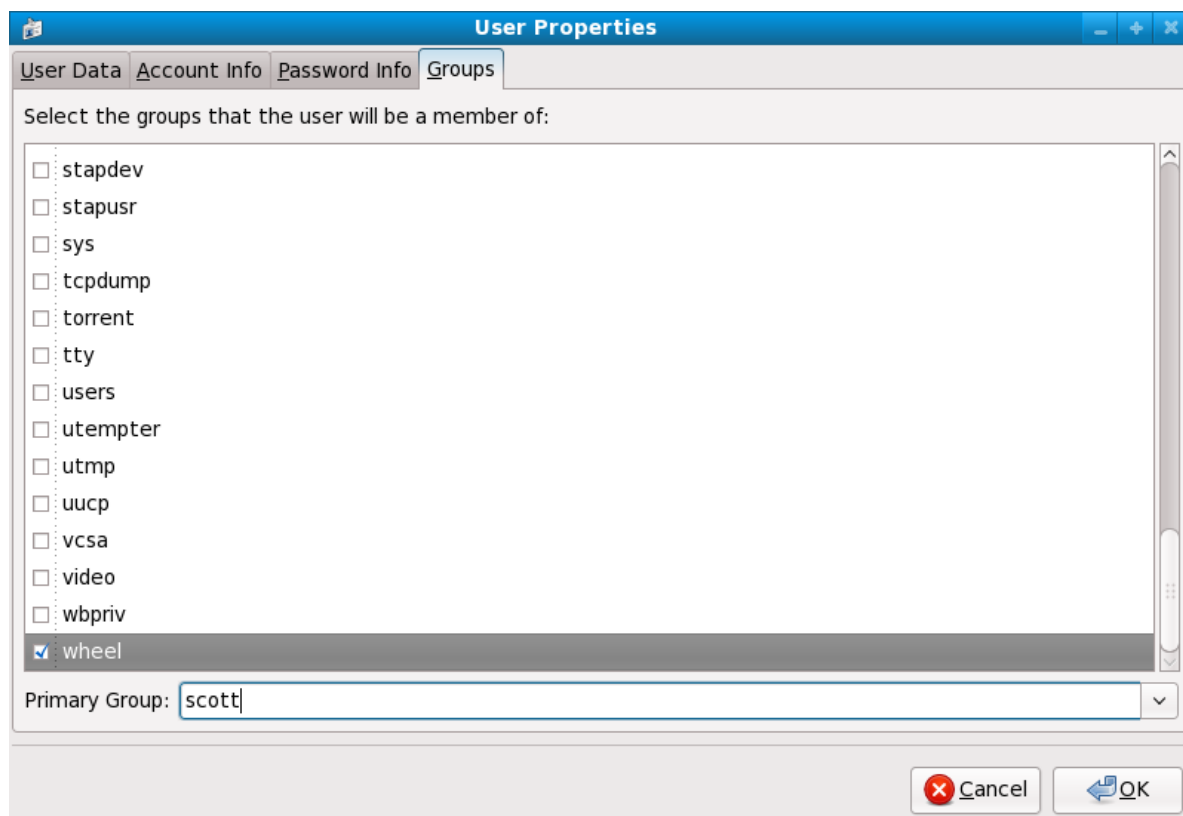
Figura 2.2. Aggiungere utenti al gruppo "wheel"

Nota

Per impostazione predefinita, l'utente root fa parte del gruppo **`wheel`**.

2.1.4.3.2. Il comando sudo

Anche il comando **sudo**, come il precedente, consente agli utenti di ottenere i privilegi amministrativi. Anteponendo **sudo** ad un comando amministrativo, viene richiesto di inserire la *propria* password. In tal modo, dopo autenticazione positiva, viene eseguito il comando come se fosse eseguito dall'utente root.

Il formato base del comando **sudo**, è il seguente:

```
sudo <command>
```

Nell'esempio precedente, <*command*> è il comando amministrativo da eseguire, per esempio il comando **`mount`**.

Importante

Gli utenti che usano il comando **sudo**, dovrebbero prestare particolare attenzione a chiudere la sessione prima di allontanarsi dalla propria macchina, giacchè tutti i sudoers (ossia gli utenti abilitati ad usare il comando sudo), possono continuare ad usare il comando per un periodo di cinque minuti, senza che venga richiesto di inserire la password. Questa impostazione può essere modificata nel file di configurazione relativo, **/etc/sudoers**.

Il comando **sudo** consente una maggiore flessibilità. Per esempio, soltanto gli utenti elencati nel file di configurazione **/etc/sudoers**, possono utilizzare il comando **sudo** che esegue nella shell dell'*utente* e non nella shell di root. Ciò significa che la sheel di root può essere completamente disabilitata. (*Sezione 2.1.4.2.1, «Disabilitare la shell di root»*).

Il comando **sudo** offre anche una registrazione degli accessi effettuati. Ogni tentativo di autenticazione è registrato nel file **/var/log/messages**, mentre il comando associato insieme allo username dell'utente è registrato nel file **/var/log/secure**.

Un altro vantaggio del comando **sudo**, deriva dal fatto che un amministratore può autorizzare gli utenti ad accedere solo a specifici comandi, secondo le loro necessità.

Per modificare il file di configurazione **/etc/sudoers** del comando **sudo**, si dovrebbe usare il comando **visudo**.

Per estendere a qualcuno pieni privilegi amministrativi, digitare **visudo** ed aggiungere, nella sezione che specifica i privilegi utenti, una riga simile alla seguente:

```
juan ALL=(ALL) ALL
```

Questo esempio stabilisce che l'utente **juan** può usare il comando **sudo** da ogni host ed eseguire ogni comando.

L'esempio seguente illustra il grado di configurazione del comando **sudo**:

```
%users localhost=/sbin/shutdown -h now
```

L'esempio stabilisce che tutti gli utenti possono lanciare il comando **/sbin/shutdown -h now**.

Le pagine di man su **sudoers** descrivono tutte le opzioni di configurazione possibili.

2.1.5. Servizi di rete disponibili

Se il controllo degli utenti sugli accessi amministrativi è un problema importante soprattutto per chi gestisce una organizzazione, monitorare quali servizi di rete devono essere attivi è di fondamentale importanza per chiunque amministri o operi con un sistema Linux.

Molti servizi in Fedora si comportano come dei server di rete. Se un servizio di rete è in esecuzione su una macchina, allora l'applicazione server (o *demone*) è in ascolto, in attesa di connessioni su una o più porte di rete. Ognuno di questi server dovrebbe essere trattato come una possbile via di attacco.

2.1.5.1. I rischi per i servizi

I servizi di rete possono creare molti rischi ai sistemi Linux. Di seguito si riporta un elenco dei principali problemi:

- *Denial of Service Attacks (DoS)* — Un attacco che intasa un servizio con raffiche di richieste, rendendo il sistema inutilizzabile.
- *Distributed Denial of Service Attack (DDoS)* — Un attacco di tipo DoS che usa più macchine compromesse (spesso in numero di mille e più), per condurre un attacco coordinato su un servizio, inondando la macchina vittima con raffiche di richieste in modo da renderla inutilizzabile.
- *Attacchi alle vulnerabilità di script* — Se un server utilizza script per eseguire compiti sul lato server, come comunemente fanno i server Web, un cracker può tentare un attacco sfruttando le vulnerabilità presenti negli script. Gli attacchi alle vulnerabilità di script, possono causare condizioni di buffer overflow o addirittura consentire l'alterazione di file.
- *Attacchi di Buffer Overflow* — I servizi che si connettono usando le porte numerate tra 0 e 1023 devono eseguire con privilegi di root, quindi se il servizio viene compromesso da un Buffer Overflow, l'attacker in ascolto può accedere al sistema con pieni privilegi. Poichè di tanto in tanto, si verificano buffer overflow nei sistemi, i cracker, per identificare i sistemi con tale vulnerabilità usano strumenti automatizzati, e una volta ottenuto l'accesso, utilizzano strumenti di rootkit automatizzati per preservare i privilegi di accesso. (n.d.t.: rootkit = accesso di livello amministrativo).

Nota

Le minacce alle vulnerabilità di tipo buffer overflow sono ridotte in Fedora, grazie a *ExecShield*, una tecnologia supportata nei kernel per mono- e multi-processori x86-compatibili che proteggono e segmentano la memoria. ExecShield riduce il rischio di buffer overflow, separando la memoria virtuale in segmenti eseguibili e non eseguibili. Ogni pezzo di programma che tenti di eseguire al di fuori del segmento eseguibile (come fanno i codici maliziosi generati da un buffer overflow), genera un segmentation fault e viene arrestato.

Execshield include supporto anche per la tecnologia *No eXecute* (NX) su piattaforme AMD64 e la tecnologia *eXecute Disable* (XD) su sistemi Itanium e Intel® 64. Queste tecnologie operano in congiunzione con ExecShield, prevenendo l'esecuzione di codice malizioso nella zone eseguibile della memoria virtuale, con una granularità di 4KB per codice.

Importante

Per limitare la possibilità di attacchi, tutti i servizi non utilizzati dovrebbero essere disattivati.

2.1.5.2. Identificare e configurare i servizi

Per aumentare la sicurezza, molti servizi di rete installati con Fedora sono disattivati per impostazione predefinita. Esistono tuttavia alcune importanti eccezioni:

- **`cupsd`** — Il server di stampa predefinito di Fedora.
- **`lpd`** — Un server di stampa alternativo.
- **`xinetd`** — Un server particolare che controlla le connessioni da alcuni server subordinati, come **`gssftp`** e **`telnet`**.

- **sendmail** — Il *Mail Transport Agent* (MTA o server di posta), sendmail, è abilitato per impostazione predefinita, ma è in ascolto solo per connessioni da localhost.

- **sshd** — Il server OpenSSH, un sicuro sostitutivo di Telnet.

In caso di indecisione se lasciare attivi questi servizi, si consiglia buon senso ed eccesso di prudenza. Per esempio, se una stampante non è disponibile, non conviene lasciare **cupsd** in esecuzione. Analogamente con **portmap**: se non si montano volumi NFSv3 o non si usa NIS (il servizio **ypbind**), allora anche il servizio **portmap** dovrebbe essere disabilitato.
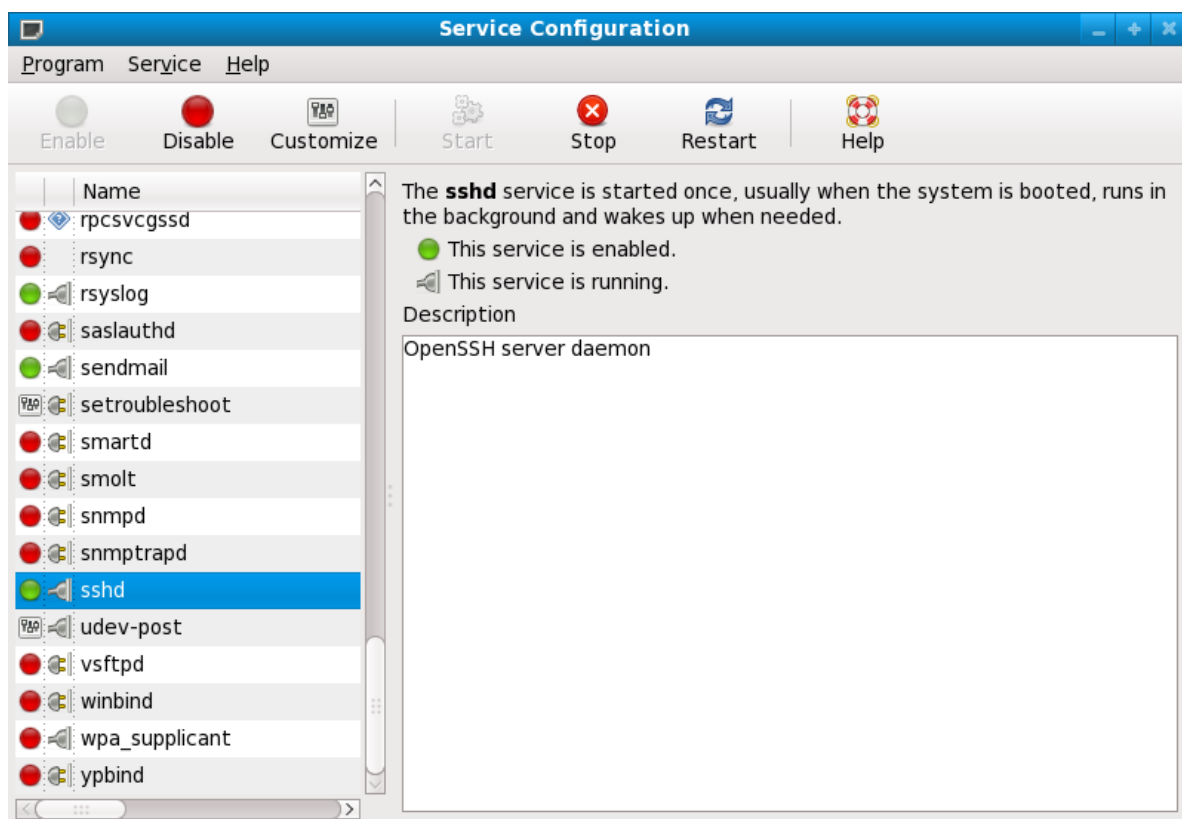
Figura 2.3. **Strumento per configurare i servizi**

Se non si è sicuri sulla funzione di un certo servizio, lo **Strumento per configurare i servizi** ha un campo descrittivo, illustrato in *Figura 2.3, «**Strumento per configurare i servizi**»*, che fornisce qualche informazione.

Controllare i servizi di rete che sia avviano al boot, costituisce soltanto una parte della storia; si dovrebbero controllare anche le porte in ascolto (o aperte). Fare riferimento alla *Sezione 2.2.8, «Controllare le porte in ascolto»*, per maggiori informazioni.

2.1.5.3. Servizi poco sicuri

Potenzialmente, tutti i servizi di rete sono poco sicuri, per questo è molto importante disabilitare i servizi non utilizzati. Falle nei servizi, vengono di tanto in tanto scoperti e corretti, per cui diventa assolutamente indispensabile aggiornare regolarmente i pacchetti associati ai servizi di rete. Vedere la *Sezione 1.5, «Aggiornamenti di sicurezza»*, per maggiori informazioni.

Alcuni protocolli di rete sono intrinsecamente molto più insicuri di altri. Tra questi servizi rientrano quelli che:

- *Trasmettono in chiaro, username e password* — Molti protocolli, piuttosto datati, come Telnet ed FTP, non cifrano la fase di autenticazione di una sessione, per cui dovrebbero essere evitati.

- *Trasmettere in chiaro, dati sensibili* — Molti protocolli trasmettono in chiaro (ossia senza cifratura), i dati. Tra questi protocolli rientrano Telnet, FTP, HTTP, ed SMTP. Anche molti file system di rete, come NFS ed SMB, trasmettono in chiaro. Usando questi protocolli, è responsabilità dell'utente stabilire se è il caso di trasmettere in chiaro i propri dati.

  Servizi remoti di memory dump, come **netdump**, trasmettono in chiaro il contenuto della memoria. Le memory dump possono contenere password, o anche i dati di un database ed altre informazioni sensibili.

  Altri servizi come **finger** e **rwhod** rivelano informazioni sugli utenti di un sistema.

Esempi di servizi intrinsecamente poco sicuri sono **rlogin**, **rsh**, **telnet** ed **vsftpd**.

Tutti i programmi shell e di accesso remoto (**rlogin**, **rsh**, e **telnet**) dovrebbero essere evitati a favore di SSH. Fare riferimento alla *Sezione 2.1.7, «Strumenti di comunicazione che aumentano la sicurezza»*, per maggiori informazioni su **sshd**.

FTP non è così inerentemente rischioso come le shell remote, tuttavia richiede configurazioni e controlli molto scrupolosi. Vedere la *Sezione 2.2.6, «Proteggere FTP»*, per maggiori informazioni sui server FTP.

I servizi che andrebbero attentamente configurati e protetti da firewall, sono:

- **finger**
- **authd** (denominato **identd** in precedenti versioni di Fedora)
- **netdump**
- **netdump-server**
- **nfs**
- **rwhod**
- **sendmail**
- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

Per maggiori informazioni su come rendere sicuri i servizi di rete, consultare la *Sezione 2.2, «Server Security»*.

Il paragrafo successivo illustra gli strumenti disponibili per impostare un semplice firewall.

2.1.6. Firewall personali

Dopo aver configurato i *necessari* servizi di rete, è importante implementare un firewall.

Importante

La configurazione dei servizi e l'implementazione di un firewall, sono operazioni da fare *prima* di connettersi ad Internet o altra rete non fidata.

Il firewall, impedisce ai pacchetti di accedere all'interfaccia di rete del sistema. Se una porta è bloccata dal firewall, ogni richiesta diretta alla porta viene ignorata. Se un servizio è in ascolto su una porta bloccata, il servizio non riceverà alcun pacchetto e di fatto risulta disabilitato. Per questo motivo, occorre prestare particolare attenzione alla configurazione di un firewall, bloccando le porte non utilizzate e sbloccando le porte dei servizi usati.

Per la maggior parte degli utenti, il miglior strumento per configurare un semplice firewall rimane l'interfaccia grafica distribuita in Fedora: **Amministrazione Firewall** (**`system-config-firewall`**). Questo strumento crea regole **`iptables`** per un firewall generico, usando una GUI.

Per maggiori informazioni sull'uso di questa applicazione e sulle opzioni disponibili, per creare un firewall di base, vedere la *Sezione 2.7.2, «Configurazione di un firewall di base»*.

Per gli utenti avanzati e gli amministratori di server, la configurazione manuale di un firewall con **`iptables`** è probabilmente una scelta migliore. Fare riferimento alla *Sezione 2.7, «Firewall»*, per maggiori informazioni. Per una guida omnicomprensiva sul comando **`iptables`**, vedere la *Sezione 2.8, «IPTables»*.

2.1.7. Strumenti di comunicazione che aumentano la sicurezza

Man mano che è aumentata la dimensione e la popolarità di Internet, è aumentata anche la minaccia delle intercettazioni. Di conseguenza, nel corso degli anni, sono stati sviluppati diversi strumenti per cifrare le comunicazioni.

Fedora, per proteggere le informazioni, distribuisce due strumenti che usano algoritmi di cifratura di alto livello e che si basano su sistemi di criptazione a chiave pubblica.

- *OpenSSH* — Una implementazione free del protocollo di comunicazione cifrata, SSH.
- *Gnu Privacy Guard (GPG)* — Una implementazione free dell'applicazione di cifratura PGP (Pretty Good Privacy).

OpenSSH, sostituendo vecchi servizi privi di cifratura come **`telnet`** e **`rsh`**, offre accessi più sicuri verso macchine remote. OpenSSH include un servizio di rete denominato **`sshd`** e tre applicazioni client da terminale:

- **`ssh`** — Una console per accesso remoto.
- **`scp`** — Un comando per copiare da/verso remoto
- **`sftp`** — Un client pseudo-ftp sicuro, per il trasferimento di file.

Per maggiori informazioni su OpenSSH, fare riferimento alla *Sezione 3.2.2, «Secure Shell»*.

Importante

Sebbene il servizio **sshd** sia inerentemente sicuro, il servizio *deve* essere tenuto aggiornato. Per maggiori informazioni, vedere la *Sezione 1.5, «Aggiornamenti di sicurezza»*.

GPG è un sistema usato anche per cifrare le e-mail. Può essere usato sia per trasmettere e-mail contenenti dati sensibili sia per cifrare i dati sensibili nei dischi.

2.2. Server Security

Quando un sistema è impiegato come un server su una rete pubblica, esso diventa un potenziale obbiettivo degli attaccanti. Consolidare il sistema e bloccare i servizi non necessari sono le operazioni che ogni buon amministratore deve effettuare.

Di seguito si riassumono alcuni utili suggerimenti di validità generale:

- Mantenere tutti i servizi aggiornati
- Usare protocolli sicuri (per quanto possibile)
- Offrire soltanto un tipo di servizio per macchina (per quanto possibile)
- Controllare attentamente tutti i servizi alla ricerca di attività sospette

2.2.1. Proteggere i servizi con TCP Wrapper e xinetd

TCP Wrapper offrono controllo d'accesso ad una varietà di servizi. La maggior parte dei servizi di rete come SSH, Telent ed FTP usano TCP Wrapper che si interpongono a guardia tra una richiesta di servizio e il servizio stesso.

I vantaggi offerti dai TCP Wrapper aumentano se usati in congiunzione con **xinetd**, un super server che garantisce ulteriore controllo su accessi, logging, binding, redirection e utilizzo delle risorse.

Nota

E' una buona idea usare anche regole di firewall, iptable, per creare ridondanza nell'ambito dei controlli d'accesso. Per maggiori informazioni sull'implementazione di firewall con i comandi iptable, fare riferimento alla *Sezione 2.7, «Firewall»*.

Di seguito si illustrano alcune opzioni di sicurezza di base.

2.2.1.1. Aumentare la sicurezza con TCP Wrapper

TCP Wrapper non solo negano l'accesso ai servizi. Questa sezione mostra come usare i TCP Wrapper per trasmettere connection banner, avvisi d'attacco da parte di host e migliorare le funzionalità di log. Per maggiori informazioni sui TCP Wrappers ed il corrispondente linguaggio, fare riferimento alle pagine man relative a **hosts_options**.

2.2.1.1.1. TCP Wrapper e Connection Banner

La visualizzazione di un banner durante la connessione ad un servizio, può rivelarsi un buon deterrente nei confronti di potenziali attaccanti, in quanto segnala la vigilanza dell'amministratore. Si possono anche selezionare le informazioni di sistema da pubblicare. Per implementare un banner TCP Wrapper per un servizio, usare l'opzione **banner**.

L'esempio implementa un banner per il servizio **vsftpd**. Iniziare, creando un file banner. Esso può essere salvato in una directory qualunque, l'importante è che abbia lo stesso nome del servizio. Per l'esempio, il file è **/etc/banners/vsftpd** con il seguente contenuto:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being removed.
```

Il token **%c** presenta una varietà di informazioni sul client, come il nome utente e l'hostname o il nome utente e l'indirizzo IP, per rendere la connessione abbastanza intimidatoria.

Per visualizzare il banner sulle richieste in corso, aggiungere la seguente riga al file **/etc/hosts.allow**:

```
vsftpd : ALL : banners /etc/banners/
```

2.2.1.1.2. TCP Wrapper e avvisi di attacco

Nel caso si siano scoperti uno o più host condurre un attacco contro il server, i TCP Wrapper possono essere configurati in modo da avvisare l'amministratore in caso di attacchi successivi, usando la direttiva **spawn**.

Di seguito si assume che un cracker dalla rete 206.182.68.0/24 stia tentando un attacco. Per impedire ogni connessione dalla rete incriminata e salvare i log dei tentativi di attacco in un file speciale, inserire la riga seguente nel file **/etc/hosts.deny**:

```
ALL : 206.182.68.0 : spawn /bin/ 'date' %c %d >> /var/log/intruder_alert
```

Il token **%d** indica il nome del servizio obbiettivo dell'attacco.

Per consentire la connessione, inserire la direttiva **spawn** nel file **/etc/hosts.allow**.

Nota

Poichè la direttiva **spawn** esegue anche comandi di shell, è una buona regola creare un particolare script che avvisi l'amministratore o che esegua una serie di comandi, ogniqualvolta un particolare client tenta di connettersi al server.

2.2.1.1.3. TCP Wrapper e messaggi di log

Se occorre tenere traccia di certe particolari connessioni, il livello di log del servizio corrispondente può essere elevato usando l'opzione **severity**.

In questo esempio si assume che chiunque tenti di connettersi alla porta 23 (la porta Telnet) di un server FTP, debba essere considerato un potenziale cracker. Per questa situazione, sostituire il flag **info** con **emerg** nel file di log, e vietare la connessione.

Inserire quindi la seguente linea nel file **/etc/hosts.deny**:

```
in.telnetd : ALL : severity emerg
```

In questo caso si usa la SyslogFacility **authpriv**, elevando la priorità dal valore predefinito **info** a **emerg**, che invia i messaggi di log direttamente alla console.

2.2.1.2. Aumentare la sicurezza con xinetd

Questa sezione spiega come usare **xinetd** per impostare un *trap service* e per controllare i livelli di risorse disponibili per un servizio. Limitare le risorse ai servizi può contribuire a contrastare gli attacchi DoS (*Denial of Service*). Fare riferimento alle pagine di man relative a **xinetd** e **xinetd.conf**, per una lista di opzioni disponibili.

2.2.1.2.1. Impostare un Trap

Una caratteristica importante di **xinetd** è la possibilità di inserire gli host, cui si vuole negare l'accesso ai servizi, in una lista **nera**. Agli host della lista è vietato, per un certo periodo di tempo o fino al successivo riavvio di **xinetd**, di accedere ai servizi gestiti da **xinetd**. Per fare ciò, occorre usare l'attributo **SENSOR**. Si tratta di un modo semplice per bloccare gli host che scansionano le porte del server.

Il primo passo da fare per impostare un **SENSOR**, è scegliere un servizio che si presume non venga utilizzato. Per questo esempio si fa riferimento a Telnet.

Nel file **/etc/xinetd.d/telnet** modificare la riga **flags** come indicato di seguito:

```
flags           = SENSOR
```

Aggiungere la seguente riga:

```
deny_time       = 30
```

L'impostazione vieta ogni tentativo di connessione verso la porta, per trenta minuti. Altri possibili valori per l'attributo **deny_time** sono FOREVER e NEVER. Il primo mantiene il divieto fino al successivo riavvio di **xinetd**; il secondo permette la connessione senza alcun divieto.

Infine, l'ultima riga:

```
disable         = no
```

L'impostazione abilita il trap.

Anche se l'utilizzo di **SENSOR** è un buon metodo per rilevare e bloccare le connessioni da host indesiderati, esso presenta due svantaggi:

- Esso non funziona nel caso di scansioni nascoste.
- Un attaccante che scopra un **SENSOR** in esecuzione, potrebbe avviare un attacco DoS contro altri host fidati e, falsificando i loro indirizzi IP, connettersi alla porta.

2.2.1.2.2. Controllare le risorse server

Un'altra importante caratteristica di **xinetd** è la sua capacità di limitare le risorse dei servizi controllati.

Per fare ciò usare le seguenti direttive:

- **`cps = <number_of_connections> <wait_period>`** — Limita il tasso di connessioni, specificando:
  - **`<number_of_connections>`** — Il numero di connessioni per secondo da gestire. Se il tasso di connessioni supera questo valore, il servizio viene temporaneamente disabilitato. Il valore predefinito è 50.
  - **`<wait_period>`** — Dopo una disabilitazione, il tempo di attesa, in secondi, prima di ri-abilitare il servizio. Il valore predefinito è 10.
- **`instances = <number_of_connections>`** — Specifica il numero totale di connessioni consentite ad un servizio. La direttiva accetta sia un valore intero sia **`UNLIMITED`**.
- **`per_source = <number_of_connections>`** — Specifica per ciascun host, il numero di connessioni consentite ad un servizio. La direttiva accetta sia un valore intero sia **`UNLIMITED`**.
- **`rlimit_as = <number[K|M]>`** — Specifica la quantità di memoria che il servizio può occupare in KB o MB. La direttiva accetta sia un valore intero sia **`UNLIMITED`**.
- **`rlimit_cpu = <number_of_seconds>`** — Specifica il periodo in secondi, dedicato al servizio dalla CPU. La direttiva accetta sia un valore intero sia **`UNLIMITED`**.

Attraverso queste direttive si può prevenire che un singolo servizio, controllato da **`xinetd`**, possa sovraccaricare il sistema, causando un DoS.

2.2.2. Proteggere Portmap

Il servizio **`portmap`** è un demone di assegnamento dinamico di porte per servizi RPC, come NIS e NFS. Può assegnare un esteso range di porte, ma presenta un meccanismo di autenticazione piuttosto debole e perciò è piuttosto difficile da rendere sicuro.

Nota

L'implementazione di una policy di sicurezza in **`portmap`** è indispensabile solo con le versioni v2 e v3 di NFS, giacchè la versione v4 non fa più uso di **`portmap`**. Se si ha intenzione di implementare un server NFSv2 o NFSv3, allora occorre usare **`portmap`** e seguire le seguenti indicazioni.

Se si eseguono servizi RPC, seguire le seguenti regole di base.

2.2.2.1. Proteggere portmap con TCP Wrapper

Data la sua mancanza di una forma di autenticazione integrata, per limitare l'accesso di reti ed host al servizio **`portmap`**, è importante usare TCP Wrapper.

Inoltre, per limitare l'accesso al servizio, usare *soltanto* indirizzi IP. Evitare di usare hostname, giacchè essi possono venir contraffatti da DNS fasulli e da altri metodi.

2.2.2.2. Proteggere portmap con iptables

Per ulteriormente restringere l'accesso al servizio **`portmap`**, è una buona idea aggiungere regole iptables al server e restringere l'accesso a reti specifiche.

Di seguito si riportano due comandi iptables. Il primo consente connessioni TCP dalla rete 192.168.0.0/24 alla porta 111 (usata dal servizio **portmap**). Il secondo consente connessioni TCP da localhost (necessario al servizio **sgi_fam** usato da **Nautilus**), alla stessa porta. Tutti gli altri pacchetti vengono scartati.

```
iptables -A INPUT -p tcp -s! 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s 127.0.0.1  --dport 111 -j ACCEPT
```

Analogamente, per limitare il traffico UDP, usare il comando:

```
iptables -A INPUT -p udp -s! 192.168.0.0/24  --dport 111 -j DROP
```

Nota

Per maggiori informazioni sull'implementazione di firewall con comandi iptables, fare riferimento alla *Sezione 2.7, «Firewall»*.

2.2.3. Proteggere NIS

NIS o Network Information Service, è un servizio RPC denominato **ypserv**, usato insieme a **portmap** e ad altri servizi per distribuire username, password ed altre informazioni sensibili agli host registrati nel dominio.

Un server NIS è costituito da varie applicazioni. Esse sono:

- **/usr/sbin/rpc.yppasswdd** — Denominato servizio **yppasswdd**, questo demone permette agli utenti di modificare la propria passowrd NIS.
- **/usr/sbin/rpc.ypxfrd** — Denominato servizio **ypxfrd**, questo demone è responsabile del trasferimento delle informazioni sensibili NIS nella rete.
- **/usr/sbin/yppush** — Questa applicazione propaga le modifiche apportate nei database NIS ai server NIS.
- **/usr/sbin/ypserv** — E' il demone del server NIS.

Secondo gli attuali standard di sicurezza, NIS è sostanzialmente poco sicuro. Esso non presenta alcun meccanismo di autenticazione degli host, trasmettendo tutte le informazioni senza alcuna cifratura, incluse le password hash. Di conseguenza, si richiede estrema attenzione alla configurazione di una rete che usi NIS. Come se non bastasse, ciò è ulteriormente complicato da una configurazione predefinita di NIS inerentemente poco sicura.

Si raccomanda quindi, a chiunque voglia implementare un server NIS, di rendere prima di tutto sicuro il servizio **portmap**, come indicato nella *Sezione 2.2.2, «Proteggere Portmap»*, e successivamente risolvere al meglio i seguenti problemi, come la pianificazione della rete.

2.2.3.1. Pianificare attentamente la rete

Poichè NIS trasmette informazioni sensibili senza usare alcuna cifratura, è importante che il servizio esegua dietro un firewall e su una rete segmentata e fidata. Se tali informazioni si trovano a transitare su una rete non fidata, essi sono a rischio di intercettazione. Una progettazione attenta della rete può aiutare a prevenire falle irrimediabili di sicurezza.

2.2.3.2. Usare una Password come Nome Dominio e Hostname

Se l'utente conosce il nome di dominio e il nome di DNS del server NIS, ogni macchina del dominio NIS può ottenere, con opportuni comandi, informazioni dal server senza bisogno di autenticazione.

Per esempio, se un utente connette un portatile alla rete o riesce ad accedere alla rete dall'esterno (ed a manomettere (spoof) un indirizzo IP interno), con il seguente comando potrebbe rivelare il contenuto del file **/etc/passwd**:

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

Se l'attaccante è in grado di accedere come root, potrebbe ottenere il file **/etc/shadow** con il comando:

```
ypcat -d <NIS_domain> -h <DNS_hostname> shadow
```

Nota

Se si usa Kerberos, il file **/etc/shadow** non è salvato in un NIS.

Per rendere più arduo ad un attaccante, l'accesso alle informazioni NIS, creare una stringa random per l'hostnome del DNS, come **o7hfawtgmhwg.domain.com** ed analogamente per il nome di dominio NIS, usando una stringa differente.

2.2.3.3. Modificare il file /var/yp/securenets

Se il file **/var/yp/securenets** è vuoto o non esiste (come capita dopo una installazione predefinita), NIS è in ascolto su tutte le reti. Quindi, una delle prime operazioni da fare è di inserire nel file, coppie di netmask/network, in modo che **ypserv** risponda solo alle richieste provenienti dalle reti specificate.

Di seguito si riporta un esempio da un file **/var/yp/securenets**:

```
255.255.255.0     192.168.0.0
```

Attenzione

Non avviare mai un server NIS senza prima aver creato un file **/var/yp/securenets** adeguato.

Questa tecnica, tuttavia, non offre protezione da un attacco di tipo IP spoofing, ma serve a limitare le reti servite da NIS.

2.2.3.4. Assegnare porte statiche ed usare regole iptables

A tutti i servizi NIS si possono assegnare porte specifiche, ad eccezione di **rpc.yppasswdd** — il demone che permette agli utenti di modificare le password di accesso. Assegnando porte ai

due demoni NIS, **rpc.ypxfrd** e **ypserv**, si possono creare regole di firewall, per proteggere ulteriormente i demoni NIS da potenziali intrusori.

Per fare ciò, aggiungere la seguenti righe al file **/etc/sysconfig/network**:

```
YPSERV_ARGS="-p 834" YPXFRD_ARGS="-p 835"
```

Per rinforzare la sicurezza, si possono poi essere usate le seguenti regole di iptables, che specificano le porte e la rete su cui il server resta in ascolto:

```
iptables -A INPUT -p ALL -s! 192.168.0.0/24  --dport 834 -j DROP
iptables -A INPUT -p ALL -s! 192.168.0.0/24  --dport 835 -j DROP
```

Con queste impostazioni, il server, a prescindere dal protocollo, accetta connessioni sulle porte 834 e 835 solo dalla rete 192.168.0.0/24.

Nota

Per maggiori informazioni sull'implementazione di firewall con comandi iptables, fare riferimento alla *Sezione 2.7, «Firewall»*.

2.2.3.5. Usare autenticazioni Kerberos

La cosa importante da considerare quando si usa NIS per autenticazione, è che ogni volta che un utente accede ad una macchina, la password hash dal file **/etc/shadow** è trasmessa in chiaro sulla rete. Se un intrusore riesce ad intrufolarsi nel dominio NIS e ad intercettare il traffico di rete, egli potrebbe carpire username e password hash. In un tempo ragionevole, un programma di crack di password potrebbe indovinare password deboli e l'attaccante ottenere un valido account d'accesso.

Poichè Kerberos usa chiavi cifrate, le password hash non sono mai trasmesse sulla rete, rendendo il sistema molto più sicuro. Per maggiori informazioni su Kerberos, vedere la *Sezione 2.6, «Kerberos»*.

2.2.4. Proteggere NFS

Importante

La versione NFSv4 inclusa in Fedora, non richiede più il servizio **portmap**, come illustrato nella *Sezione 2.2.2, «Proteggere Portmap»*. In tutte le versioni di NFS, il traffico viene trasmesso usando TCP e non più UDP. Inoltre NFSv4 ora include autenticazioni utente e di gruppo basati su Kerberos, parte integrante del modulo **RPCSEC_GSS** del kernel. Si includono informazioni anche su **portmap**, giacchè Fedora supporta sia NFSv2 sia NFSv3 che utilizzano **portmap**.

2.2.4.1. Pianificare attentamente la rete

Ora che NFSv4 usa Keberos per trasmettere le informazioni (cifrate), è importante che il servizio venga correttamente configurato dietro un firewall o su una porzione di rete. NFSv2 ed NFSv3 continuano a trasmettre i dati in chiaro e di ciò va tenuto conto. Una accurata progettazione di rete, che tenga conto di ciò, aiuta a prevenire falle di sicurezza.

2.2.4.2. Attenzione agli errori sintattici

Il server NFS determina i file system da esportare e verso quali host, consultando il file **/etc/exports**. Prestare molta attenzione a non aggiungere spazi durante la modifica del file.

Per esempio, la seguente riga nel file **/etc/exports**, condivide la directory **/tmp/nfs/** con l'host **bob.example.com** con permessi read/write.

```
/tmp/nfs/     bob.example.com(rw)
```

Invece a causa dello spazio dopo l'hostname, la seguente riga nel file **/etc/exports**, condivide la directory con l'host **bob.example.com** in sola lettura, e la condivide con *tutti gli altri* in lettura/scrittura.

```
/tmp/nfs/     bob.example.com (rw)
```

E' una buona norma verificare ogni condivisione NFS configurata, usando il comando **showmount**:

```
showmount -e <hostname>
```

2.2.4.3. Non usare l'opzione no_root_squash

Per impostazione, le condivisioni NFS modificano l'utente root nell'utente **nfsnobody**, un account utente senza privilegi. Il risultato è che il proprietario di tutti i file creati da root diventa **nfsnobody**, impedendo l'avvio di programmi setuid.

Se si usa l'opzione **no_root_squash**, un utente root remoto può modificare ogni file nel sistema condiviso e lasciare applicazioni malevoli, come trojan, che potrebbero essere inavvertitamente eseguiti da ignari utenti.

2.2.4.4. Configurazione di firewall in NFS

Le porte usate da NFS sono assegnate dinamicamente da rpcbind, che potrebbe causare problemi durante la creazione delle regole di firewall. Per semplificare il processo, usare il file */etc/sysconfig/nfs* per specificare le porte da usare:

- **MOUNTD_PORT** — Porta TCP e UDP per mountd (rpc.mountd)
- **STATD_PORT** — Porta TCP e UDP per lo stato (rpc.statd)
- **LOCKD_TCPPORT** — Porta TCP per nlockmgr (rpc.lockd)
- **LOCKD_UDPPORT** — Porta UDP per nlockmgr (rpc.lockd)

I numeri di porta specificati non devono essere usati da altri servizi. Configurare il firewall per autorizzare le porte specificate, insieme alla porte UDP e TCP 2049 (NFS).

Usare il comando **rpcinfo -p** sul server NFS per vedere le porte e i programmi RPC usati.

2.2.5. Proteggere HTTP Apache

Il server HTTP Apache è uno dei servizi più stabili e sicuri distribuiti con Fedora. Un gran numero di opzioni e tecniche sono disponibili per rendere sicuro il server HTTP Apache — troppe per essere analizzate tutte quì con la necessaria dovizia. La seguente sezione spiega brevemente, buone pratiche di utilizzo del server HTTP Apache.

Verificare sempre che gli script in esecuzione sul sistema funzionino correttamente, *prima* di renderli effettivi in sistemi di produzione. Inoltre, assicurarsi che soltanto l'utente root abbia permessi di scrittura nelle directory contenente script o CGI. Per fare ciò eseguire i seguenti comandi, come root:

1. ```
   chown root <directory_name>
   ```

2. ```
   chmod 755 <directory_name>
   ```

Gli amministratori di sistema dovrebbero prestare la massima attenzione nell'uso delle seguenti direttive, configurabili in **`/etc/httpd/conf/httpd.conf`**:

`FollowSymLinks`
: La direttiva è abilitata per impostazione; prestare la dovuta attenzione a non creare link simbolici al root document del server web. Per esempio, sarebbe una pessima idea creare un link simbolico a `/`.

`Indexes`
: La direttiva è abilitata per impostazione, ma potrebbe non essere desiderabile. Per impedire ai visitatori di sfogliare i file sul server, disabilitare questa direttiva.

`UserDir`
: La direttiva **`UserDir`**, è disabilitata per impostazione perchè può confermare la presenza di un account nel sistema. Per consentire la visualizzazione della directory di un utente, usare le seguenti direttive:

```
UserDir enabled
UserDir disabled root
```

Queste direttive consentono la navigazione nelle directory degli utenti, esclusa la directory **`/root/`**. Per aggiungere altre directory da disabilitare, aggiungere gli account utenti, separati da spazio, alla riga **`UserDir disabled`**.

Importante

Non rimuovere la direttiva **`IncludesNoExec`**. Per impostazione, il modulo *SSI* (Server-Side Includes) non può eseguire comandi. Si raccomanda di non cambiare questa impostazione a meno che non sia strettamente necessario, poichè potrebbe abilitare un attaccante ad eseguire comandi.

2.2.6. Proteggere FTP

FTP (File Transfer Protocol) è un vetusto protocollo TCP progettato per il trasferimento di file. Poichè tutte le transazioni con il server, inclusa l'autenticazione, sono in chiaro, FTP è considerato un protocollo non sicuro e perciò richiede opportune configurazioni.

Fedora offre tre server FTP

- **`gssftpd`** — Un demone FTP che non trasmette informazioni di autenticazioni, basato su **`xinetd`** e controllato da Kerberos.
- **Red Hat Content Accelerator** (**`tux`**) — Un server web dello spazio kernel con capacità FTP.

- **vsftpd** — Un servizio FTP a sè stante orientato alla sicurezza.

Di seguito si indicano le linee guida per impostare un servizio FTP, **vsftpd**.

2.2.6.1. Greeting Banner FTP

Prima di inviare le proprie credenziali di accesso (username e password), gli utenti vengono salutati con un banner di benvenuto. Per impostazione, il banner include informazioni sulla versione usata, che potrebbero essere maliziosamente usate da un cracker, note le vulnerabilità di sistema.

Per modificare le impostazioni del banner, aggiungere la seguente direttiva al file **/etc/vsftpd/vsftpd.conf**:

```
ftpd_banner=<insert_greeting_here>
```

Sostituire <*insert_greeting_here*> nella direttiva precedente con il messaggio di benvenuto.

Per banner su più righe, conviene usare un file banner. Per semplificare la gestione di banner multipli, posizionare tutti i banner in una directory denominata **/etc/banners/**. In questo esempio, il file banner per connessioni FTP è **/etc/banners/ftp.msg**. Ecco un esempio di file banner:

```
######### # Hello, all activity on ftp.example.com is logged. #########
```

Nota

Come specificato nella *Sezione 2.2.1.1.1, «TCP Wrapper e Connection Banner»*, non occorre iniziare ogni riga del file con **220**.

Per fare riferimento a questo file banner, aggiungere la seguente direttiva al file **/etc/vsftpd/vsftpd.conf**:

```
banner_file=/etc/banners/ftp.msg
```

Usando i TCP Wrapper, come descritto nella *Sezione 2.2.1.1.1, «TCP Wrapper e Connection Banner»*, è possibile inviare ulteriori banner alle connessioni in entrata.

2.2.6.2. Accesso anonimo

La directory **/var/ftp/** attiva l'account anonimo.

Il modo più semplice per creare la directory è di installare il pacchetto **vsftpd**. Il pacchetto crea una directory per utenti anonimi e configura in sola lettura la directory.

Per impostazione, gli utenti anonimi non possono scrivere in nessuna directory.

Attenzione

Se si abilita l'accesso anonimo al server FTP, prestare attenzione a dove sono salvati i dati sensibili.

2.2.6.2.1. Upload anonimo

Per consentire ad utenti anonimi di inviare file sul server, si raccomanda di creare una directory in sola scrittura in **/var/ftp/pub/**.

Ecco la procedura; digitare il comando:

```
mkdir /var/ftp/pub/upload
```

Poi, modificare i permessi in modo che gli utenti anonimi non possano vedere (o sfogliare) il contenuto della directory:

```
chmod 730 /var/ftp/pub/upload
```

Un listato *long format* della directory apparirebbe così:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

Attenzione

Gli amministratori che permettono ad utenti anonimi di leggere e scrivere in directory, spesso scoprono che i loro server diventano repository di software pirata.

Poi, aggiungere la seguente riga al file **/etc/vsftpd/vsftpd.conf**:

```
anon_upload_enable=YES
```

2.2.6.3. Account utenti

Poichè FTP trasmette username e password in chiaro, è una buona norma vietare agli utenti l'accesso al server, con i loro account.

Per disabilitare tutti gli account, aggiungere la seguente direttiva al file **/etc/vsftpd/vsftpd.conf**:

```
local_enable=NO
```

2.2.6.3.1. Restringere gli account utenti

Per disabilitare gli accessi FTP ad utenti o gruppi specifici, come l'utente root e quelli con privilegi **sudo**, si può usare un file di autenticazione PAM, come descritto nella *Sezione 2.1.4.2.4, «Disabilitare l'account root usando PAM»*. Il file di configurazione PAM relativo a **vsftpd** è **/etc/pam.d/vsftpd**.

E' anche possibile disabilitare gli account direttamente all'interno di ciascun servizio.

Per disabilitare un account specifico, aggiungere lo username nel fie **/etc/vsftpd.ftpusers**.

2.2.6.4. Usare TCP Wrapper per il controllo degli accessi

Consultare la *Sezione 2.2.1.1, «Aumentare la sicurezza con TCP Wrapper»*, per controllare gli accessi al servizio FTP usando TCP Wrapper.

2.2.7. Proteggere Sendmail

Sendmail è un MTA (Mail Transfer Agent) che usa SMTP (Simple Mail Transfer Protocol) per trasferire posta elettronica tra altri MTA ed ai client di posta. Sebbene molti MTA siano capaci di cifrare le comunicazioni, la maggior parte di essi non lo sono, perciò spedire posta elettronica su una rete pubblica è considerato una forma di comunicazione inerentemente non sicura.

A chiunque sia desideroso di implemetare un server Sendmail, si raccomanda di seguire le seguenti indicazioni.

2.2.7.1. Limitare un attacco tipo DoS

Data la natura dei messaggi di posta elettronica, un attaccante potrebbe molto facilmente sovraccaricare il server inondandolo con flussi ininterrotti di messaggi (fllooding), causando un Denial of Service (DoS). Impostando i limiti alle seguenti direttive, presenti nel file **/etc/mail/sendmail.mc**, si limita il rischio legato a tali attacchi.

- **confCONNECTION_RATE_THROTTLE** — Il numero di connessioni al secondo accettate dal server. Per impostazione, Sendmail non presenta un limite al numero di connessioni. Se viene impostato un limite ed esso viene superato, le future connessioni vengono ritardate.
- **confMAX_DAEMON_CHILDREN** — Il numero massimo di processi (child) generati dal processo server (parent). Per impostazione, Sendmail non assegna alcun limite al numero di processi child. Se viene impostato un limite e superato, le future connessioni vengono ritardate.
- **confMIN_FREE_BLOCKS** — Il numero minimo di blocchi che devono rimanere liberi perchè il server continui a ricevere mail. Il valore predefinito è 100.
- **confMAX_HEADERS_LENGTH** — La dimensione massima, in byte, per l'intestazione (header) del messaggio.
- **confMAX_MESSAGE_SIZE** — La dimensione massima, in byte, per un singolo messaggio.

2.2.7.2. NFS e Sendamil

Non porre mai la directory di coda delle mail, **/var/spool/mail/** su un volume condiviso NFS.

Poichè NFSv2 ed NFSv3 non usano alcun controllo sugli ID degli utenti e dei gruppi, due o più utenti potrebbero risultare con lo stesso ID, e ricevere e leggere le mail reciproche.

Nota

Con NFSv4 che usa Kerberos, questo non è il caso, in quanto il modulo **`SECRPC_GSS`** del kernel, non fa uso di autenticazioni basate su ID. Comunque rimane valida la considerazione di *non* porre la directory di coda delle mail su volumi condivisi NFS.

2.2.7.3. Utenti di sola posta elettronica

Per impedire che utenti locali possano attaccare il server Sendmail, sarebbe meglio limitare l'accesso al server solo tramite un programma di posta. Gli account di shell sul mail server non dovrebbero essere permessi e tutte le shell degli utenti, nel file **`/etc/passwd`**, dovrebbero essere impostate su **`/sbin/nologin`** (con la possibile eccezione dell'utente root).

2.2.8. Controllare le porte in ascolto

Dopo aver configurato i servizi di rete, diventa di primaria importanza prestare la dovuta attenzione alle porte effettivamente in ascolto sulle interfacce di rete. Ogni porta aperta è un rischio di intrusione.

Esistono due approcci di base per elencare le porte in ascolto. Quello meno affidabile è interrogare lo stack di rete usando comandi come **`netstat -an`** o **`lsof -i`**. Il metodo è poco affidabile, in quanto questi programmi non si connettono alla macchina dalla rete, ma controllano i servizi in esecuzione sul sistema. Per questo motivo, queste applicazioni sono frequenti obbiettivi degli attaccanti. I cracker, in genere, nascondono le tracce dei loro interventi sulle porte che sono riusciti ad aprire, sostituendo **`netstat`** e **`lsof`** con proprie versioni modificate.

Un metodo più affidabile per controllare le porte aperte, è usare uno scanner come **`nmap`**.

Il seguente comando digitato in un terminale, determina le porte in ascolto su connessioni TCP:

```
nmap -sT -O localhost
```

L'uscita del comando assomiglia a:

```
Starting Nmap 5.21 ( http://nmap.org ) at 2010-07-08 19:00 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00016s latency).
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.32.14-127.fc12.i686.PAE
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.44 seconds
```

L'output mostra i servizi in esecuzione. Nell'esempio, un sospetto potrebbe venire sul servizio *unknown* in esecuzione sulla porta TCP 834. Per controllare se le porta è associata alla lista ufficiale dei servizi noti, si lancia il comando:

```
cat /etc/services | grep 834
```

Nel caso dell'esempio non si ha alcun output. Quindi, nonostante la porta faccia parte di un range di porte riservate (0 - 1023), e la sua apertura richiederebbe il permesso di root, essa non è associata ad alcun servizio noto.

Allora, si prova ad ottenere alcune informazioni sulla porta, usando il comando **netstat** o **lsof**. Per controllare la porta 834 con **netstat**, si digita:

```
netstat -anp | grep 834
```

Il comando restituisce il seguente output:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

La scoperta fatta con **netstat** che la porta è aperta, è abbastanza rassicurante, poichè un cracker che apra furtivamente una porta non ne permetterebbe la rivelazione con questo comando. Inoltre, l'opzione **[p]** rivela l'ID del processo (PID) che ha aperto la porta. In questo caso la porta appartiene a **ypbind** (NIS) che è un servizio RPC gestito insieme al servizio **portmap**.

L'uscita del comando **lsof** sarebbe molto simile al precedente, giacchè anch'esso è in grado di collegare le porte aperte ai servizi:

```
lsof -i | grep 834
```

La porzione di output rilevante per il nostro esempio è:

```
ypbind      653        0    7u  IPv4       1319                 TCP *:834 (LISTEN)
ypbind      655        0    7u  IPv4       1319                 TCP *:834 (LISTEN)
ypbind      656        0    7u  IPv4       1319                 TCP *:834 (LISTEN)
ypbind      657        0    7u  IPv4       1319                 TCP *:834 (LISTEN)
```

Questi strumenti rivelano una grande quantità di informazioni sullo stato dei servizi in esecuzione. Essi sono flessibili ed offrono una varietà di informazioni sui servizi e la configurazione di rete. Per maggiori informazioni vedere le pagine di man relative a **lsof**, **netstat**, **nmap**, e **services**.

2.3. Single Sign-on (SSO)

2.3.1. Introduzione

La funzionalità SSO di Fedora serve a ridurre il numero di autenticazioni richieste agli utenti Fedora. La maggior parte delle applicazioni sfruttano gli stessi meccanismi di autenticazione ed autorizzazione, cosicchè una volta loggati in Fedora, gli utenti non devono reinserire la loro password. Queste applicazioni sono illustrate più avanti.

Inoltre, gli utenti possono accedere alle loro macchine anche in assenza di una connessione di rete (*modalità offline*), oppure in condizioni di connessioni inaffidabili, per esempio in accessi wireless. In quest'ultimo caso, il livello dei servizi risulterà leggermente degradato.

2.3.1.1. Applicazioni supportate

Di seguito si elencano le applicazioni che attualmente supportano lo schema di accesso unificato in Fedora:

- Login
- Salvaschermo
- Firefox e Thunderbird

2.3.1.2. Meccanismi di autenticazione supportati

Fedora correntemente supporta i seguenti meccanismi di autenticazione:

- Login via nome/password Kerberos
- Login via Smart Card

2.3.1.3. Smart Card supportate

Fedora è stato testato con il lettore e le smart-card Cyberflex, ma anche altre smart-card conformi alle specifiche Java card 2.1.1 e Global Platform 2.0.1 dovrebbero operare correttamente, come ogni lettore che sia supportato dalla piattaforma PCSC.

Fedora è stato testato anche con lo standard Common Access Cards (CAC) (n.d.t. impiegato principalmente negli U.S.A. dal DoD). Il lettore supportato per CAC è l'SCM SCR 331 USB.

Fedora supporta anche smart card Gemalto Cyberflex Access 64k v2, conformi con gli standard DER SHA-1 configurati come in PKCSI v2.1. Queste smart card ora usano lettori che si conformano alle norme CCID (Chip/Smart Card Interface Devices).

2.3.1.4. Vantaggi di Single Sign-on di Fedora

Oggigiorno, esistono numerosi meccanismi di sicurezza che utilizzano una varietà di protocolli e di *credential store*. Tra questi si ricordano SSL, SSH, IPsec e Kerberos. L'SSO di Fedora si propone di unificare questi schemi. Ciò non vuol dire sostituire Keberos con certificazioni X.509v3, quanto unificarli in modo da ridurre il carico di gestione sia agli utenti che agli amministratori.

Per raggiungere questo obbiettivo Fedora:

- Presenta, in ogni sistema operativo, una singola istanza condivisa delle librerie di criptazione NSS.
- Include il Sistema di Certficazione ESC (Enterprise Security Client), con il sistema operativo base. L'applicazione ESC intercetta gli eventi relativi all'inserzione delle samrt card. Se una smart card, conforme al Sistema di Certificazione usato in Fedora viene inserita nel sistema, ESC visualizza una interfaccia grafica istruendo l'utente su come registrare la smart card.
- Unifica Kerberos e NSS in modo che gli utenti che accedono al sistema usando una smart card, possano ottenere anche una credenziale Kerberos (in modo da poter accedere a file server ed altri servizi).

2.3.2. Primo utilizzo di una nuova Smart Card

Prima di poter usare la smart card sul proprio sistema e avvantaggiarsi delle possibilità di sicurezza offerte da questa tecnologia, occorre effettuare alcune installazioni e configurazioni, come descritto di seguito.

Nota

Questo paragrafo offre una descrizione generale su come iniziare ad usare la propria smart card. Per informazioni più dettagliate consultare "Red Hat Certificate System Enterprise Security Client Guide".

1. Accedere con le proprie credenziali (nome/password) Kerberos.

2. Assicurarsi che sia installato il pacchetto **`nss-tools`**.

3. Scaricare ed installare i propri certificati. Usare il seguente comando per installare il root CA certificate:

```
certutil -A -d /etc/pki/nssdb -n "root ca cert" -t "CT,C,C" -i ./
ca_cert_in_base64_format.crt
```

4. Verificare che siano installati i seguenti pacchetti: esc, pam_pkcs11, coolkey, ifd-egate, ccid, gdm, authconfig, ed authconfig-gtk.

5. Abilitare l'accesso via Smart Card

   a. Nel menu di GNOME, selezionare Sistema->Amministrazione->Autenticazione.

   b. Inserire, quando richiesto, la password di root.

   c. Nella finestra di Configurazione dell'Autenticazione, selezionare la scheda **Autenticazione**.

   d. Spuntare la checkbox **Abilitare il supporto per Smart Card**.

   e. Cliccare sul bottone **Configura Smart Card...** per modificare le impostazioni di Smartcard:

      - **Richiedere smart card, per accedere** — Disabilitare la checkbox. Una volta effettuato l'accesso con la smart card, si può abilitare questa opzione per impedire l'accesso senza una smart card.

      - **In caso di rimozione** — Una volta effettuato l'accesso, questa opzione imposta alcuni eventi legati alla rimozione della smart card. Le opzioni possibili sono:

        - **Blocca** — La rimozione della smart card provoca il blocco dello schermo.

        - **Ignora** — La rimozione della smart card non provoca alcun effetto.

6. Se occorre abilitare OCSP (Online Certificate Status Protocol), aprire il file **`/etc/pam_pkcs11/pam_pkcs11.conf`** e individuare la riga contenente la seguente opzione:

   `enable_ocsp = false;`

   Modificare come indicato di seguito:

   `enable_ocsp = true;`

7. Registrare la smart card

8. Se si usa una card CAC, occorre completare i seguenti passaggi:

    a. Come utente root, creare un file denominato **/etc/pam_pkcs11/cn_map**.

    b. Al file **cn_map** appena creato, aggiungere la riga seguente:

        MY.CAC_CN.123454 -> *myloginid*

        dove, *MY.CAC_CN.123454* è il Common Name sulla propria card CAC e *myloginid* è il proprio UID di accesso.

9. Logout

2.3.2.1. Risoluzione problemi

In caso di problemi con la smart card, per localizzare la causa del problema provare ad usare il seguente comando (smart card registrata ed inserita nel lettore):

```
pklogin_finder debug
```

Il comando **pklogin_finder** in modalità debug, cerca di recuperare la validità dei certificati e di verificare se uno UID sia associato ad uno dei certificati presenti nella card.

2.3.3. Come funziona la registrazione di una Smart Card

Le smart card vengono *registrate* nel momento in cui ricevono un certificato firmato da un CA (Autorità di Certificazione). Il processo involve diversi passaggi, descritti di seguito:

1. L'utente inserisce la propria smart card in un lettore nei pressi della macchina. Questo evento è intercettato da ESC (Enterprise Security Client).

2. Sul desktop dell'utente viene visualizzata la pagina di registrazione. L'utente inserisce le necessarie informazioni, dopodichè il sistema contatta il TPS (Token Processing System) e il CA.

3. Il TPS registra la smart card usando un certificato firmato dal CA.
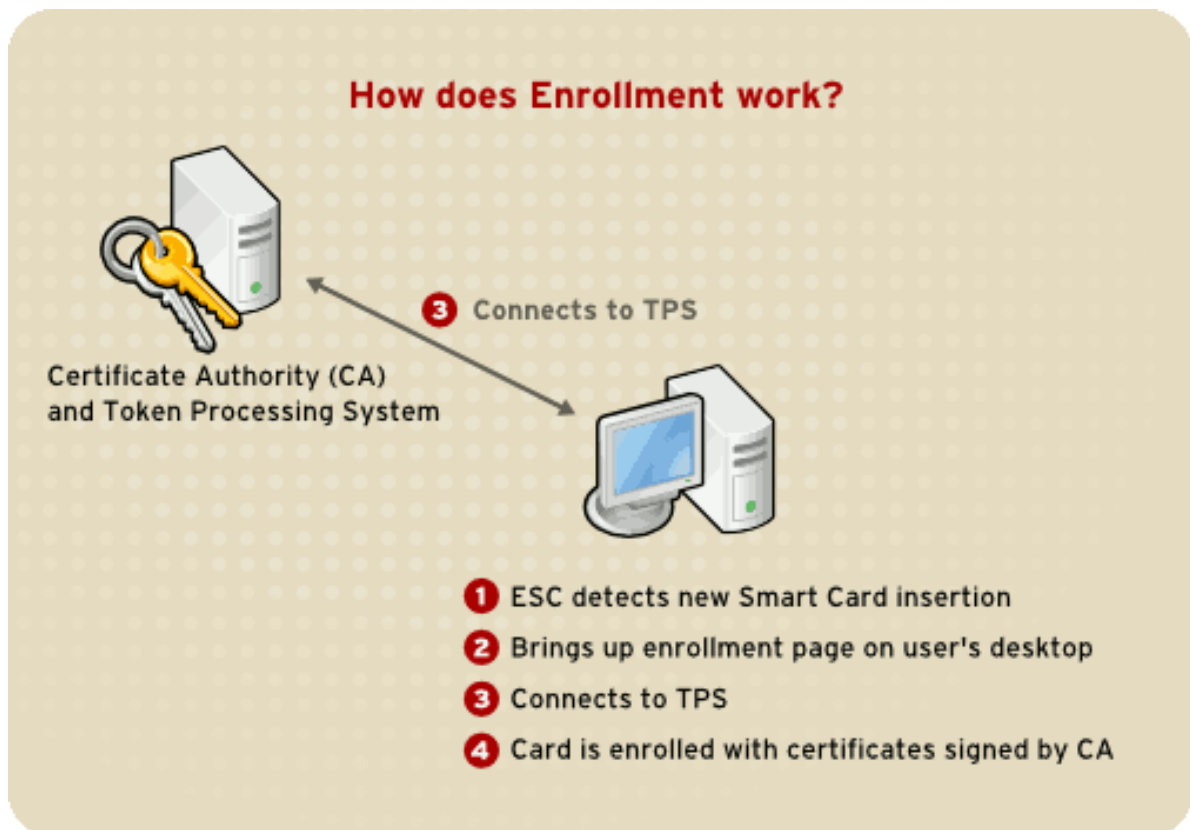
Figura 2.4. Come funziona la registrazione di una Smart Card

2.3.4. Come funziona l'accesso via Smart Card

Questo paragrafo offre una breve panoramica sul processo di accesso usando smart card.

1. Quando l'utente inserisce la propria smart card nel lettore, l'evento è intercettato da PAM che chiede di inserire il PIN utente.

2. Quindi, il sistema controlla i certificati attuali dell'utente e verifica la loro validità. Il certificato è successivamente associato all'UID dell'utente.

3. Infine il KDC conferma e autorizza l'accesso.
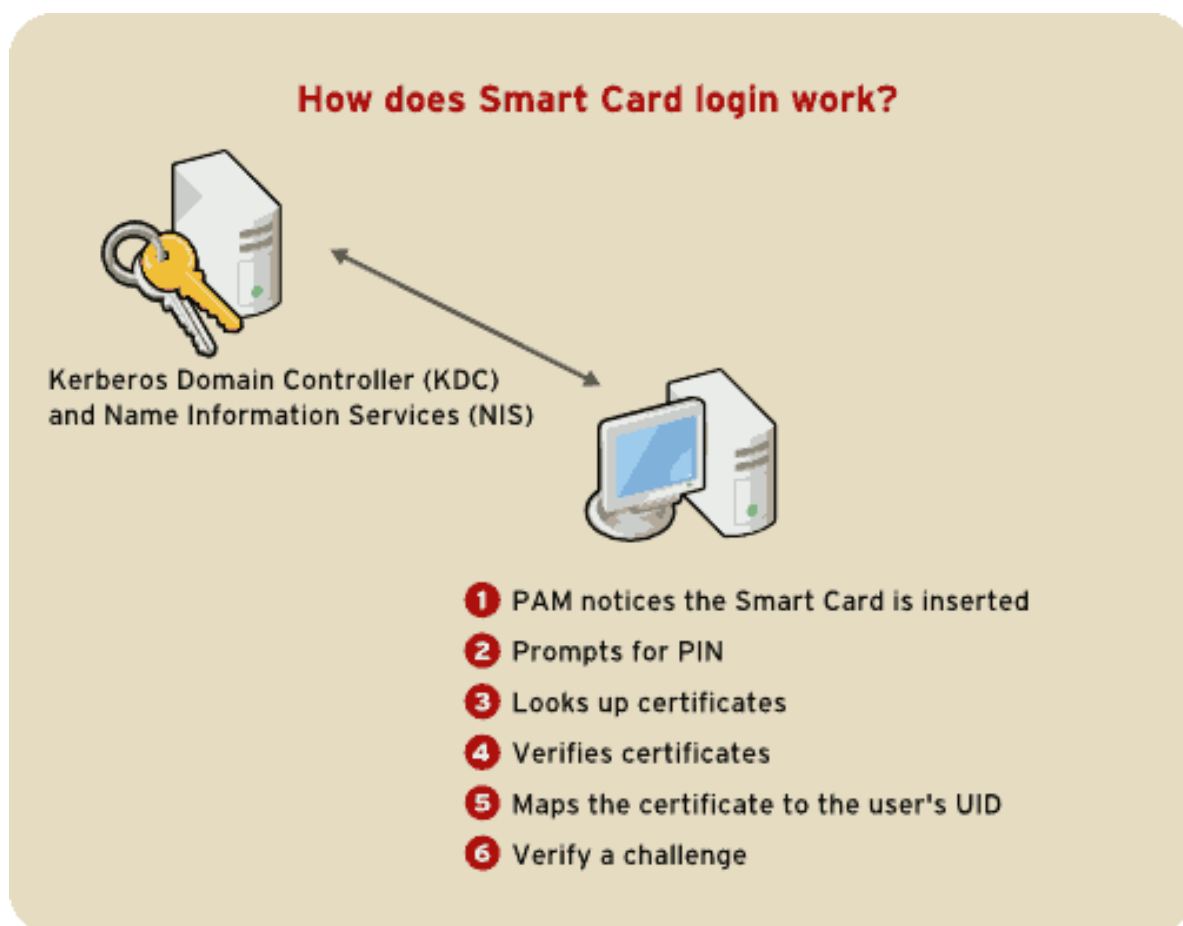
Figura 2.5. Come funziona l'accesso via Smart Card

Nota

Non è possibile accedere al sistema con una card non registrata anche se formattata: per accedere al sistema, occorre possedere una card che sia formattata e registrata.

Vedere la *Sezione 2.6, «Kerberos»* e la *Sezione 2.4, «Pluggable Authentication Modules (PAM)»*, per maggiori informazioni su Kerberos e PAM.

2.3.5. Configurare Firefox ad usare Kerberos con SSO

E' possibile configurare Firefox ad usare Kerberos con SSO. Perchè questa funzionalità operi correttamente, occorre configurare il browser in modo da inviare le credenziali Kerberos al KDC appropriato. Il seguente paragrafo descriverà i passi necessari per una corretta configurazione.

1. Per visualizzare le attuali opzioni di configurazione, nella barra degli indirizzi di Firefox digitare **`about:config`**.

2. Nel campo **Filter**, digitare **`negotiate`** per restringere la lista delle opzioni.

3. Fare doppio click sull'opzione *network.negotiate-auth.trusted-uris*, per visualizzare la finestra di dialogo *Inserimento stringa*.

4. Inserire il nome del dominio entro cui si richiede di essere autenticati, per esempio *`example.com`*.

5. Ripetere i passi precedenti con il campo *network.negotiate-auth.delegation-uris*, usando lo stesso nome di dominio.

Nota

Si può lasciare vuoto questo campo, giacchè autorizza il passaggio dei ticket Kerberos, che non è richiesto.

Se queste due opzioni di configurazione non sono elencate, si sta usando una versione di Firefox troppo vecchia, per cui si consiglia di effettuare un up-grade.
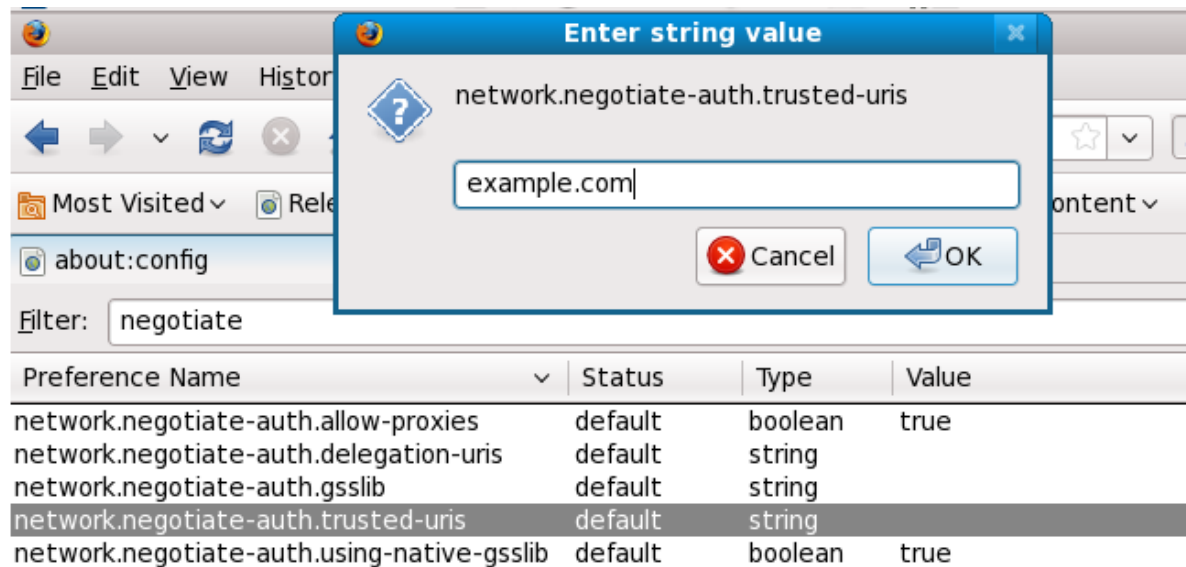
Figura 2.6. Configurazione di Firefox per SSO con Kerberos

A questo punto occorre assicurarsi di avere i ticket Kerberos. In un terminale, digitare **`kinit`** per recuperare i ticket. Per visualizzare la lista dei ticket disponibili, digitare **`klist`**. Di seguito si mostra un esempio di utilizzo di questi comandi:

```
[user@host ~] $ kinit
Password for user@EXAMPLE.COM:

[user@host ~] $ klist
Ticket cache: FILE:/tmp/krb5cc_10920
Default principal: user@EXAMPLE.COM

Valid starting     Expires            Service principal
10/26/06 23:47:54  10/27/06 09:47:54  krbtgt/USER.COM@USER.COM
        renew until 10/26/06 23:47:54

Kerberos 4 ticket cache: /tmp/tkt10920
klist: You have no tickets cached
```

2.3.5.1. Risoluzione problemi

Se si sono seguiti i passaggi di configurazione indicati ma il processo di autenticazione non funziona, è possibile attivare in modalità verbosa, i messaggi del processo di autenticazione. In tal modo è possibile individuare la causa del problema. Per abilitare la modalità verbosa, seguire i seguenti passaggi:

1. Chiudere tutte le istanze di Firefox.

2. Aprire un terminale e digitare i seguenti comandi:

   ```
   export NSPR_LOG_MODULES=negotiateauth:5
   export NSPR_LOG_FILE=/tmp/moz.log
   ```

3. Riavviare Firefox *dal terminale* e visitare il sito che precedentemente dava problemi di autenticazione. I vari messaggi saranno registrati in **`/tmp/moz.log`**, dove una loro analisi potrà fornire una soluzione al problema. Per esempio:

   ```
   -1208550944[90039d0]: entering nsNegotiateAuth::GetNextToken()
   -1208550944[90039d0]: gss_init_sec_context() failed: Miscellaneous failure
   No credentials cache found
   ```

   Nel caso sovraindicato non si hanno i ticket Kerberos, per cui occorre eseguire **`kinit`**.

Se **`kinit`** esegue con successo sulla propria macchina, ma l'autenticazione non riesce, allora nel file di log comparirà qualcosa del genere:

```
-1208994096[8d683d8]: entering nsAuthGSSAPI::GetNextToken()
-1208994096[8d683d8]: gss_init_sec_context() failed: Miscellaneous failure
Server not found in Kerberos database
```

Generalmente ciò indica un problema di configurazione di Kerberos. Assicurarsi che, siano esatte, le impostazioni nella sezione [domain_realm] del file **`/etc/krb5.conf`**. Per esempio:

```
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Se il file di log è vuoto, probabilmente si è dietro un proxy, il quale elimina le intestazioni HTTP necessarie per il processo di autenticazione. Un modo per raggirare il problema, consiste nel connettersi al server usando HTTPS, che permette alla richiesta di passare senza modificazioni. Quindi procedere alla fase di debug, ricorrendo come suggerito al file di log.

2.4. Pluggable Authentication Modules (PAM)

I programmi che autorizzano l'accesso ad un sistema, usano l'*autenticazione* per verificare l'identità degli utenti (autenticazione, vuol dire, stabilire che un utente è chi dice di essere).

Nel passato, ogni programmi aveva un proprio modo per autenticare gli utenti. Con Fedora molti programmi sono stati configurati per usare un meccanismo di autenticazione centralizzato, denominato PAM (Pluggable Authentication Modules).

PAM presenta un architettura modulare, offrendo all'amministratore un alto grado di flessibilità per impostare le policy di autenticazione nel sistema.

Nella maggior parte dei casi, il file di configurazione predefinito risulta pressochè sufficiente per una applicazione che usa PAM. Altre volte, risulta invece necessario editare un file PAM di configurazione.

Poichè errori di configurazione possono compromettere la sicurezza del sistema, è importante capire la strutture di questi file prima di apportare qualsiasi modifica. Per maggiori informazioni, fare riferimento alla *Sezione 2.4.3, «Formato del file di configurazione di PAM»*.

2.4.1. Vantaggi di PAM

PAM presenta i seguenti vantaggi:

- uno schema di autenticazione comune che può essere usato in un'ampia varietà di applicazioni.
- significativa flessibilità e controllo sull'autenticazione, sia per gli amministratori sia per gli sviluppatori di applicazioni.
- una singola libreria completamente documentata, che permette agli sviluppatori di scrivere programmi senza bisogno di creare i propri schemi di autenticazione.

2.4.2. File di configurazione di PAM

La directory **`/etc/pam.d/`** contiene i file di configurazione di PAM di ciascuna applicazione che usa PAM. Nelle precedenti versioni di PAM veniva usato il file **`/etc/pam.conf`**, ora deprecato ed usato unicamente su sistemi che non hanno la directory **`/etc/pam.d/`**.

2.4.2.1. File PAM del servizio

Ogni applicazione o *servizio* che usi PAM, possiede un file nella directory **`/etc/pam.d/`**. Ciascun file di questa directory ha lo stesso nome del servizio di cui controlla l'accesso.

Un programma che usa PAM è responsabile di definire il nome del servizio e di installare il proprio file di configurazione PAM nella directory **`/etc/pam.d/`**. Per esempio il programma **`login`** definisce il suo nome di servizio come **`login`** e installa il proprio file di configurazione PAM **`/etc/pam.d/login`**.

2.4.3. Formato del file di configurazione di PAM

Ogni file di configurazione PAM contiene un gruppo di direttive strutturate come segue:

```
<module interface>  <control flag>   <module name>   <module arguments>
```

Ciascuno di questi elementi è spiegato nelle seguenti sezioni.

2.4.3.1. Module Interface

Attaualmente sono disponibili quattro tipi di interfacce di moduli PAM. Ciascuna di esse corrisponde a un differente aspetto del processo di autenticazione:

- **`auth`** — Questa interfaccia autentica l'uso. Per esempio richiede e verifica la validità di una password. I moduli con questa interfaccia possono anche impostare credenziali, come l'appartenenza ad un gruppo o i ticket Kerberos.
- **`account`** — Questa interfaccia verifica il permesso di accesso. Per esempio controlla la scadenza di un account o controlla il permesso di accesso in una data ora del giorno.
- **`password`** — Questa interfaccia è usata per modificare la password degli utenti.
- **`session`** — Questa interfaccia configura e gestisce le sessioni. I moduli con questa interfaccia possono anche effettuare ulteriori operazioni necessarie in un accesso, come montare la home directory di un utente o rendere disponibile la casella di posta di un utente.

Nota

Un singolo modulo può presentare una o più interfacce. Per esempio **`pam_unix.so`** presenta tutte e quattro le interfacce.

In un file di configurazione di PAM, l'interfaccia è il primo campo definito. Per esempio, una tipica riga in un file di configurazione è simile a questa:

```
auth required pam_unix.so
```

Questa direttiva stabilisce di usare l'interfaccia **`auth`** del modulo **`pam_unix.so`**.

2.4.3.1.1. Impilare Module Interface

Le direttive di interfaccia possono essere *impilate*, ossia disposte una sull'altra, cosicchè più moduli possano essere usati per realizzare una certa finalità. Se il flag di controllo di un modulo ha il valore "sufficient" o "requisite" (sul significato di questi flag di controllo, fare riferimento alla *Sezione 2.4.3.2, «Control Flag»*), allora ai fini del processo di autenticazione è importante l'ordine in cui i moduli sono disposti nella lista.

La disposizione in pila permette ad un amministratore di specificare le condizioni necessarie da soddisfare, prima di avviare il processo di autenticazione. Per esempio il comando **`reboot`**, generalmente usa diversi moduli impilati, come si può vedere nel suo file di configurazione PAM:

```
[root@MyServer ~]# cat /etc/pam.d/reboot
#%PAM-1.0
auth sufficient pam_rootok.so
auth required pam_console.so
#auth include  system-auth
account required pam_permit.so
```

- La prima riga è un commento e non viene presa in considerazione.
- **`auth sufficient pam_rootok.so`** — Questa riga usa il modulo **`pam_rootok.so`** che verifica se l'utente corrente è l'utente root, controllando che il suo UID sia 0. Se il test ha successo, gli altri moduli non vengono presi in considerazione e il comando eseguito. Se il test fallisce, viene preso in considerazione il modulo successivo.
- **`auth required pam_console.so`** — Questa riga usa il modulo **`pam_console.so`** che tenta di autenticare l'utente. Se l'utente è gia loggato in un terminale, **`pam_console.so`** controlla se nella directory **`/etc/security/console.apps/`** esiste un file con lo stesso nome del servizio (reboot). Se il file esiste, l'autenticazione ha successo ed il controllo passa al modulo successivo.
- **`#auth include system-auth`** — Questa riga è un commento e perciò non processata.
- **`account required pam_permit.so`** — Questa riga usa il modulo **`pam_permit.so`** che consente all'utente root o ad altro utente loggato in un terminale di riavviare il sistema.

2.4.3.2. Control Flag

Tutti i moduli PAM quando vengono chiamati, danno un esito positivo o negativo. I flag di controllo, in base all'esito della chiamata, indicano a PAM cosa fare. I moduli possono essere impilati in un ordine

particolare ed i flag determinano quanto sia rilevante un successo o fallimento di un dato modulo, nel processo di autenticazione dell'utente.

Ci sono quattro flag di controllo predefiniti:

- **`required`** — Il risultato sul modulo deve essere positivo perchè l'autenticazione continui. Se il test fallisce in questo punto, l'utente non riceve alcuna notifica finchè non vengono completati tutti i test dei moduli che fanno riferimento all'interfaccia.
- **`requisite`** — Il risultato sul modulo deve essere positivo perchè l'autenticazione continui. Comunque, se un test fallisce in questo punto, l'utente è immediatamente notificato con un messaggio che indica il primo test di modulo **`required`** *o* **`requisite`** fallito.
- **`sufficient`** — Il risultato sul modulo viene ignorato in caso di fallimento. Inoltre, se il test di un modulo contrassegnato **`sufficient`** ha successo *e* nessun modulo precedente contrassegnato **`required`** è fallito, allora non è richiesto nessun'altro test e l'utente è autenticato per il servizio.
- **`optional`** — Il risultato sul modulo viene ignorato. Un modulo contrassegnato con **`optional`** non è rilevante per l'autenticazione, se esiste un'altra interfaccia che fa riferimento all'interfaccia stessa.

Importante

Non è critico l'ordine di chiamata dei moduli **`required`**. Soltanto i flag **`sufficient`** e **`requisite`** fanno diventare importante l'ordine.

Correntemente, è disponibile una nuova sintassi per i flag di controllo che consente un controllo più preciso su PAM.

Le pagine di man su **`pam.d`** e la documentazione su PAM nella directory **`/usr/share/doc/pam-`*`<version-number>`*`/`**, in cui <*`version-number`*> è la versione di PAM sul proprio sistema, descrivono questa nuova sintassi in tutti i dettagli.

2.4.3.3. Module Name

Il nome di un modulo consente a PAM di fare riferimento al modulo contenente la specifica interfaccia. Nelle precedenti versioni di Fedora, si usava indicare il percorso completo del modulo, nel file di configurazione di PAM. Inoltre, con la comparsa dei sistemi *multilib*, che utilizzano moduli PAM a 64 bit di **`/lib64/security/`**, il nome della directory viene omesso perchè l'applicazione è collegata alla versione **`libpam`** appropriata, in grado di localizzare la corretta versione del modulo.

2.4.3.4. Module Arguments

Durante la fase di autenticazione, PAM usa *argomenti* per passare informazioni ad un modulo.

Per esempio il modulo **`pam_userdb.so`**, usa le informazioni contenute in un file di database Berkley DB, per autenticare l'utente. Il Berkley DB è un database open source incluso in molte applicazioni. Il modulo accetta un argomento **db** che specifica il database da usare.

Di seguito si riporta una riga tipica relativa a un modulo **`pam_userdb.so`** in un file di configurazione di PAM. Il <*`path-to-file`*> rappresenta il percorso completo al file di database Berkley DB:

```
auth required pam_userdb.so db=<path-to-file>
```

Il passaggio di argomenti non validi, *generalmente* non altera il successo o fallimento della chiamata del modulo PAM. Comunque in caso di fallimento, gli errori sono riportati nel file **`/var/log/secure`**.

2.4.4. Un esempio di file di configurazione di PAM

Di seguito si riporta un esempio di file di configurazione di PAM:

```
#%PAM-1.0
auth   required  pam_securetty.so
auth   required  pam_unix.so nullok
auth   required  pam_nologin.so
account  required  pam_unix.so
password required  pam_cracklib.so retry=3
password required  pam_unix.so shadow nullok use_authtok
session required  pam_unix.so
```

- La prima riga è un commento, contrasseganta dal carattere "cancelletto" (**#**) posto all'inizio della riga.
- Le righe comprese tra la seconda e la quarta impilano tre moduli per autenticare l'accesso.

  `auth required pam_securetty.so` — Questo modulo controlla che il tty su cui l'utente si sta loggando sia presente nel file **`/etc/securetty`**, *se* l'utente tenta di accedere come root.

  Se il tty non è presente, ogni tentativo di accedere come root fallisce con un messaggio **`Login errato`**.

  `auth required pam_unix.so nullok` — Questo modulo richiede all'utente una password e poi confronta la password usando le informazioni presenti nel file **`/etc/passwd`** e se esiste, nel file **`/etc/shadow`**.

  - L'argomento **`nullok`** indica al modulo **`pam_unix.so`** di permettere l'uso di pasword vuote.
- **`auth required pam_nologin.so`** — Questo modulo controlla se esiste il file **`/etc/nologin`**. Se il file esiste e l'utente non è l'utente root, l'autenticazione fallisce.

Nota

In questo esempio, vengono controllati tutti e tre i moduli **`auth`**, anche in caso di fallimento nel primo modulo. In tale situazione l'utente non sa a quale stadio sia fallita l'autenticazione, ed anche per un attaccante diventa più gravoso capire come crackare il sistema.

- **`account required pam_unix.so`** — Questo modulo verifica l'account. Per esempio verifica se è abilitata l'illegibilità delle password e l'interfaccia account del modulo **`pam_unix.so`** controlla la scedenza dell'account o se l'utente ha modificato la password nel periodo indicato.
- **`password required pam_cracklib.so retry=3`** — Se una password è scaduta, il componente relativo al modulo **`pam_cracklib.so`** richiede di inserire una nuova password. E poi verifica che la nuova password sia abbastanza robusta.
  - L'argomento **`retry=3`** specifica che se la verifica fallisce una prima volta, l'utente ha altre due possbilità per creare una password robusta.

- **`password required pam_unix.so shadow nullok use_authtok`** — Questa riga indica che per cambiare la password utente, occorre usare l'interfaccia **`password`** del modulo **`pam_unix.so`**.
  - L'argomento **`shadow`** indica che il modulo crea password illegibili durante l'aggiornamento di una password.
  - L'argomento **`nullok`** indica che il modulo permette all'utente di cambiare la propria password da una *vuota* (una password vuota indica un account bloccato).
  - L'ultimo argomento su questa riga, **`use_authtok`**, è un esempio dell'importanza dell'ordinamento in una pila di moduli PAM. Questo argomento indica di non richiedere di inserire una nuova password. Infatti, si accetta qualsiasi password accettata da un modulo precedente. In questo caso tutte le nuove password devono superare la verifica del modulo **`pam_cracklib.so`** che garantisce password sicure.
- **`session required pam_unix.so`** — La riga finale indica all'interfaccia della sessione del modulo **`pam_unix.so`** di gestire la sessione. Questo modulo registra nel file **`/var/log/secure`** il nome utente e il tipo di servizio, all'inizio ed alla fine di ogni sessione. Questo modulo può essere integrato con altri moduli di sessione per ulteriori funzionalità.

2.4.5. Creare moduli PAM

E' possibile creare o aggiungere in ogni momento, nuovi moduli PAM alle applicazioni che usano PAM.

Per esempio, uno sviluppatore potrebbe sviluppare un metodo per generare password "usa e getta" e realizzare un modulo PAM di supporto. Poi, i programmi che usano PAM possono immediatamente usare il nuovo modulo ed il nuovo programma di generazione password, senza bisogno di ricompilazioni o di altre modifiche.

Questo consente agli sviluppatori ed agli amministratori di mescolare insieme, come pure testare metodi di autenticazione su differenti programmi, senza bisogno di ricompilazione.

La documentazione relativa alla realizzazione di moduli è inclusa nella directory **`/usr/share/doc/pam-`*`<version-number>`*`/`**, dove *`<version-number>`* è la versione di PAM in uso nel sistema.

2.4.6. Caching delle credenziali PAM ed Amministrative

In Fedora, un numero di strumenti amministrativi permette agli utenti di ottenere elevati privilegi per un periodo di cinque minuti, tramite il modulo **`pam_timestamp.so`**. E' importante capire il funzionamento di questo meccanismo, perchè un utente che si allontani da un terminale mentre **`pam_timestamp.so`** è ancora in vita, lascia la macchina aperta a manipolazioni da parte di chiunque possa fisicamente accedere al terminale incustodito.

Nello schema di temporizzazione di PAM, l'applicazione di amministrazione grafica richiede all'utente di inserire la password di root. Ad autenticazione avvenuta, il modulo **`pam_timestamp.so`** crea un file a marca temporale. Per impostazione, il file viene creato nella directory **`/var/run/sudo/`**. Se il file esiste già, l'interfaccia non richiede la password. Infatti il modulo **`pam_timestamp.so`** sovrascrive il file a marca temporale esistente, riservando altri cinque minuti di accesso amministrativo all'utente.

Si può controllare l'attuale stato del file a marca temporale, ispezionando il file **`/var/run/sudo/<user>`**. Nell'uso desktop, il file rilevante è **`unknown:root`**. Se è presente e la sua marca temporale è inferiore a cinque minuti, le credenziali sono ancora valide.

L'esistenza del file a marca temporale, è confermata da un'icona di autenticazione che appare nell'area di notifica del pannello.

Figura 2.7. L'Icona di Autenticazione

2.4.6.1. Rimuovere il file a marca temporale

Prima di lasciare incustodita una macchina in cui sia attiva una temporizzazione di PAM, si raccomanda di distruggere il file contenente la marca temporale. Per fare questo in un ambiente grafico, cliccare l'icona di autenticazione nel *system tray*.
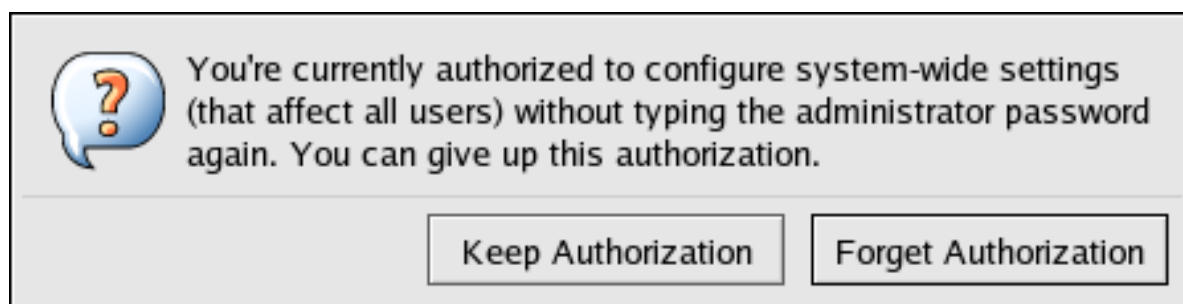
Figura 2.8. Rimuovere l'Autenticazione

Occorre prestare attenzione ai seguenti aspetti del file a marca temporale di PAM:

- Se l'accesso avviene da remoto usando **`ssh`**, usare il comando **`/sbin/pam_timestamp_check -k root`** per eliminare il file a marca temporale
- Occorre lanciare il comando **`/sbin/pam_timestamp_check -k root`** dallo stesso terminale da cui è stata avviata l'applicazione privilegiata.
- Occorre essere loggati con l'account dell'utente che ha originariamente invocato il modulo **`pam_timestamp.so`**, per poter usare il comando **`/sbin/pam_timestamp_check -k`**. Non accedere come utente root per eseguire questo comando.
- Se si vuole eliminare le credenziali sul desktop (senza usare l'cona **Dimentica Autorizzazione**), usare il seguente comando:

  ```
  /sbin/pam_timestamp_check -k root </dev/null >/dev/null 2>/dev/null
  ```

  Eventuali fallimenti del comando rimuovono soltanto le credenziali (se presenti) dal tty da cui è stato eseguito il comando.

Per maggiori informazioni sull'uso del comando **`pam_timestamp_check`**, per eliminare il file a marca temporale, fare riferimento alle pagine di man relative a **`pam_timestamp_check`**.

2.4.6.2. Comuni direttive di pam_timestamp

Il modulo **`pam_timestamp.so`** accetta diverse direttive. Le seguenti sono le due opzioni più comunemente usate:

- **`timestamp_timeout`** — Specifica il periodo di validità del file a marca temporale (in secondi). Il valore predefinito è 300 (5 minuti).
- **`timestampdir`** — Specifica la directory in cui è salvato il file a marca temporale. Il valore predefinito è **`/var/run/sudo/`**.

Vedere la *Sezione 2.7.9.1, «Documentazione installata riguardante i firewall»*, per maggiori informazioni su come gestire il modulo **`pam_timestamp.so`**.

2.4.7. Proprietario di PAM e di Dispositivo

In Fedora, il primo utente che accede al terminale della macchina, può manipolare certi dispositivi ed effettuare certe operazioni normalmente pertinenti all'utente root. Tale controllo avviene tramite un modulo di PAM, denominato **`pam_console.so`**.

2.4.7.1. Il proprietario di Dispositivo

Quando un utente accede ad un sistema Fedora, il modulo **`pam_console.so`** è chiamato da **`login`** o dal programma d'accesso grafico usato, **gdm**, **kdm** o **xdm**. Se l'utente è il primo ad accedere ad una console fisica — riferito anche come *console user* — il modulo attribuisce all'utente il diritto di proprietà su una verietà di dispositivi normalmente attrbuiti all'utente root. Il *console user* rimane il proprietario di questi dispositivi fino al termine della sua ultima sessione locale. Una volta uscito, l'utente root torna ad essere il proprietario.

I dispositivi interessati includono, ma non solo, schede audio, drive di dischetti e drive CD.

Questa possibilità permette ad un utente locale di manipolare questi dispositivi, senza bisogno di accedere come utente root, semplificando così comuni compiti al *console user*.

E' possibile modificare la lista dei dispositivi controllati dal modulo **`pam_console.so`**, modificando i seguenti file:

- **`/etc/security/console.perms`**
- **`/etc/security/console.perms.d/50-default.perms`**

Nei file indicati, si possono cambiare i permessi anche a dispositivi che non fanno parte della lista oppure si possono modificare le impostazioni predefinite. Piuttosto che modificare direttamente il file **`50-default.perms`**, si consiglia di creare un nuovo file (per esempio ***`xx`*`-name.perms`**), in cui inserire le modifiche richieste. Il nome del nuovo file predefinito, deve iniziare con un numero maggiore di 50 (per esempio, **`51-default.perms`**). In questo modo il sistema PAM non terrà conto del file predefinito **`50-default.perms`**.

Attenzione

Se il file di configurazione del gestore dello schermo, **gdm**, **kdm** o **xdm** è stato modificato per consentire l'accesso da remoto *e* l'host è configurato per eseguire al runlevel 5, allora si raccomanda di modificare le direttive **`<console>`** e **`<xconsole>`**, nel file **`/etc/security/console.perms`** con i seguenti valori:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :0\.[0-9] :0
<xconsole>=:0\.[0-9] :0
```

Ciò serve ad impedire ad utenti remoti di accedere ai dispositivi ed alle applicazioni riservate della macchina.

Se il file di configurazione del gestore dello schermo, è stato modificato per permettere l'accesso da remoto *e* l'host è stato configurato per eseguire ad un qualsiaisi runlevel multi-utente diverso da 5, si raccomanda di rimuovere completamente la direttiva **`<xconsole>`** e di modificare la direttiva **`<console>`** con il seguente valore:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]*
```

2.4.7.2. Accesso alle Applicazioni

Il *console user* ha anche accesso a certi programmi i cui utilizzi sono configurati nella directory **/etc/security/console.apps/**

Questa directory contiene i file di configurazione che abilitano il *console user* ad eseguire certe applicazioni presenti nelle directory **/sbin** e **/usr/sbin**.

Questi file di configurazione hanno lo stesso nome delle applicazioni di cui conservano le impostazioni.

Un gruppo importante di applicazioni a cui ha accesso il *console user*, è costituito da quelle applicazioni che consento di spegnere o riavviare il sistema:

- **/sbin/halt**
- **/sbin/reboot**
- **/sbin/poweroff**

Poichè queste applicazioni sono supportate da PAM, il loro utilizzo richiede che sia chiamato il modulo **pam_console.so**.

Per maggiori informazioni, fare riferimento alla *Sezione 2.7.9.1, «Documentazione installata riguardante i firewall»*.

2.4.8. Ulteriori risorse

Le seguenti risorse spiegano ulteriormente i metodi da usare per configurare PAM. In aggiunta a queste, si consiglia di investigare i file di configurazione presenti nel sistema per meglio comprendere la loro struttura.

2.4.8.1. Documentazione su PAM installata

- Pagine man relative a PAM — Sono disponibili diverse pagine di man sulle varie applicazioni e sui file di configurazione riguardanti PAM. Di seguito si riporta un elenco delle più importanti pagine di man:

  File di configurazione

  - **pam** — Una buona introduzione a PAM con una spiegazione della struttura e degli impieghi dei file di configurazione di PAM.

    Notare che questa pagina di man, descrive sia il file **/etc/pam.conf** sia i singoli file di configurazione nella directory **/etc/pam.d/**. Per impostazione, Fedora usa file di configurazione individuali, in **/etc/pam.d/**, ignorando completamente **/etc/pam.conf** (anche se presente).

  - **pam_console** — Descrive lo scopo del modulo **pam_console.so**. Descrive anche la sintassi appropriata per ogni direttiva nel file di configurazione di PAM.

  - **console.apps** — Descrive il formato e le opzioni disponibili nel file di configurazione **/etc/security/console.apps**, che specifica le applicazioni accessibili al *console user* assegnate da PAM.

  - **console.perms** — Descrive il formato e le opzioni disponibili nel file di configurazione **/etc/security/console.perms**, che specifica i permessi assegnati da PAM al *console user*.

- **pam_timestamp** — Descrive il modulo **pam_timestamp.so**.

- ***/usr/share/doc/pam-<version-number>*** — Contiene *System Administrators's Guide*, *Module Writers' Manual* e *Application Developers' Manual*, come pure una copia dello standard PAM, DCE-RFC 86.0, in cui *<version-number>* è la versione di PAM.

- ***/usr/share/doc/pam-<version-number>/txts/README.pam_timestamp*** — Contiene informazioni sul modulo **pam_timestamp.so**, in cui *<version-number>* è la versione di PAM.

2.4.8.2. Siti web utili su PAM

- *http://www.kernel.org/pub/linux/libs/pam/* — Il sito web principale del progetto Linux-PAM, con informazioni sui vari moduli di PAM, una FAQ e documenti.

Nota

La documentazione presente nel sito sopra citato, riguarda la versione di PAM più recente e potrebbe non essere conforme al 100% alla versione inclusa in Fedora.

2.5. TCP Wrapper e xinetd

Controllare l'accesso ai servizi di rete, è una delle operazioni di sicurezza più importanti che un amministratore di server deve fronteggiare. E Fedora offre diversi strumenti al riguardo. Per esempio, un firewall basato su regole **iptables** che filtra i pacchetti indesiderati, nell'ambito dello stack di rete del kernel; *TCP Wrapper* che aggiungono un ulteriore livello di protezione definendo gli host autorizzati/non autorizzati a connettersi ai servizi di rete, "*wrapped*". Un esempio di servizio *wrapped* (avvolto, coperto), è il *super server* `xinetd`. Il servizio è detto *super server* perchè controlla le connessioni in un insieme ristretto di servizi, raffinando ulteriormente il controllo d'accesso.

La *Figura 2.9, «Controllo d'accesso ai servizi di rete»* schematizza il funzionamento complessivo degli strumenti a protezione dei servizi di rete.
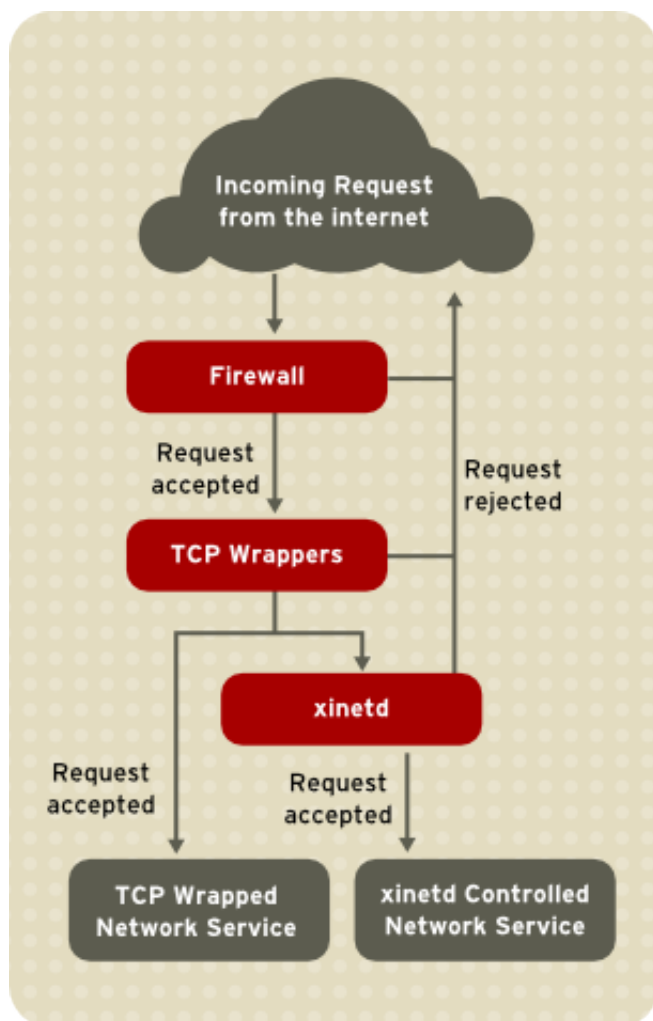
Figura 2.9. Controllo d'accesso ai servizi di rete

Questo capitolo si concentra sul ruolo dei TCP Wrapper e di `xinetd` nel controllare l'accesso ai servizi di rete e mostra come impiegare questi strumenti per migliorare sia i messaggi di log sia la gestione dei servizi controllati. Per informazioni sull'uso di firewall, con regole **`iptables`**, fare riferimento alla *Sezione 2.8, «IPTables»*.

2.5.1. TCP Wrapper

Il pacchetto TCP Wrapper (**`tcp_wrappers`**) viene installato automaticamente in ogni sistema Fedora e fornisce controlli d'accesso basati su host. Il componente principale del pacchetto è costituito dalla libreria **`libwrap.a`**. In termini generali, un servizio TCP-Wrapped è un servizio compilato usando la libreria **`libwrap.a`**

Quando si effettua una connessione ad un servizio TCP-Wrapped, il servizio dapprima fa riferimento ai file d'accesso degli host (**`/etc/hosts.allow`** e **`/etc/hosts.deny`**), verificando se il client è autorizzato a connettersi. Poi, nella maggior parte dei casi, usa il demone syslog (`syslogd`) per registrare il nome del client ed il servizio richiesto nel file **`/var/log/secure`** o **`/var/log/messages`**.

Se il client è autorizzato, TCP Wrapper rilascia il controllo della connessione al servizio, senza alcuna ulteriore interposizione nella comunicazione tra client e server.

Oltre al controllo d'accesso e al logging, TCP Wrapper durante la fase di connessione, ossia prima di negare o passare il controllo al servizio, può eseguire comandi d'interazione con il client.

Poichè i TCP Wrapper sono un valore aggiunto per l'arsenale di strumenti a disposizione di ogni amministratore, i principali servizi di rete in Fedora sono linkati alla libreria **`libwrap.a`**. Tra di essi figurano `/usr/sbin/sshd`, **`/usr/sbin/sendmail`** e `/usr/sbin/xinetd`.

Nota

Per verificare se un servizio è linkato alla libreria **`libwrap.a`**, come utente root digitare il comando:

```
ldd <binary-name> | grep libwrap
```

Sostituire <*`binary-name`*> con il nome del servizio di rete.

Se il comando restituisce un output vuoto, allora il servizio *non* è linkato.

Di seguito si riporta l'output di un servizio (`/usr/sbin/sshd`) linkato:

```
[root@myServer ~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@myServer ~]#
```

2.5.1.1. Vantaggi dei TCP Wrapper

Un TCP Wrapper fornisce i seguenti vantaggi rispetto ad altre tecniche di controllo dei servizi di rete:

- *Transparenza nei confronti sia del client sia del servizio di rete wrapped* — Sia il client sia il servizio wrapped sono inconsapevoli dell'impiego di TCP wrapper. Gli utenti legittimati vengono connessi al servizio, mentre quelli non legittimati vengono bloccati.
- *Gestione centralizzata di protocolli multipli* — I TCP Wrapper operano in maniera indipendente dai servizi e consentono, a molte applicazioni server, di condividere un insieme comune di file di configurazione di controllo d'accesso, semplificando la gestione.

2.5.2. File di configurazione di TCP Wrapper

Per determinare se un client può connettersi ad un servizio, i TCP Wrapper fanno riferimento ai seguenti due file, comunemente denominati file degli *host access*:

- **`/etc/hosts.allow`**
- **`/etc/hosts.deny`**

Quando un servizio TCP-Wrapped riceve una richiesta da un client, il sistema effettua i seguenti passaggi:

1. *Fa referimento a **`/etc/hosts.allow`**.* — Il servizio TCP-wrapped scorre in sequenza il file **`/etc/hosts.allow`**, applicando la prima regola definita per il servizio. Se esiste una regola compatibile, la connessione viene autorizzata; altrimenti continua con il passaggio successivo.
2. *Fa referimento a **`/etc/hosts.deny`**.* — Il servizio TCP-wrapped scorre in sequenza il file **`/etc/hosts.deny`**. Se esiste una regola compatibile, la connessione viene negata; altrimenti autorizza l'accesso al servizio.

Di seguito si riportano alcune importanti considerazioni sull'utilizzo dei TCP Wrapper :

- Poichè le regole di accesso elencate in **`hosts.allow`** sono applicate per prima, esse hanno la precedenza sulle regole specificate in **`hosts.deny`**. Quindi, se l'accesso ad un servizio è permesso secondo **`hosts.allow`**, una eventuale regola di divieto presente in **`hosts.deny`** viene ignorata.
- Le regole in ciascun file sono lette dalla cima verso il basso, e la prima regola trovata è l'unica che viene applicata. Quindi è rilevante l'ordine d'inserimento.
- L'accesso al servizio è garantito, se i file non esistono o se in entrambi i file non esiste alcuna regola per il servizio.
- I servizi TCP-wrapped non caricano in memoria (in cache) le regole dei file d'accesso, perciò ogni modifica apportata ai file **`hosts.allow`** o **`hosts.deny`** ha effetto immediato, senza bisogno di riavviare i servizi.

Attenzione

Se l'ultima riga di un file d'accesso non termina con un carattere di ritorno a capo (newline, ossia premendo il tasto **`Invio`**), l'ultima regola nel file fallisce restituendo un messaggio di errore in **`/var/log/messages`** e **`/var/log/secure`**. Lo stesso accade per una regola suddivisa su più righe che non terminano con il carattere backslash (\). Il seguente esempio illustra una porzione di un messaggio di log relativo ad una regola che fallisce a causa delle circostanze citate:

```
warning: /etc/hosts.allow, line 20: missing newline or line too long
```

2.5.2.1. Formattare le Regole di Accesso

Il formato è identico per entrambi i file **`/etc/hosts.allow`** e **`/etc/hosts.deny`**. Ogni regola deve trovarsi sulla propria linea. Le linee vuote o che iniziano con il carattere diesis o cancelletto (#) vengono ignorate.

Ogni regola usa il seguente formato base per controllare l'accesso ai servizi di rete:

```
<daemon list>: <client list> [: <option>: <option>: ...]
```

- *`<daemon list>`* — Un elenco di nomi di processo (*non* nomi di servizio), separati da virgole o il termine riservato **`ALL`**. L'elenco accetta anche operatori, garantendo una grande flessibilità d'utilizzo (*Sezione 2.5.2.1.4, «Operatori»*).
- *`<client list>`* — Un elenco di hostname, indirizzi IP, pattern speciali o termini riservati, separati da virgole, che identificano gli host interessati dalla regola. L'elenco accetta anche operatori (*Sezione 2.5.2.1.4, «Operatori»*).
- *`<option>`* — Un'azione opzionale o un elenco di azioni da eseguire, separate da virgole, all'intercettazione di una regola. Il campo option supporta espansioni, comandi di shell, permette/autorizza l'accesso e permette di modificare il comportamento dei messaggi di log.

Nota

Maggiori informazioni sui termini indicati, si trovano in altre sezioni di questa Guida:

- *Sezione 2.5.2.1.1, «Wildcards»*
- *Sezione 2.5.2.1.2, «Pattern»*
- *Sezione 2.5.2.2.4, «Espansioni»*
- *Sezione 2.5.2.2, «Campi Opzioni»*

Di seguito si riporta un esempio di una semplice regola d'accesso:

```
vsftpd : .example.com
```

Questa regola indica di controllare le connessioni provenienti dagli host del dominio `example.com` e dirette verso il demone FTP (`vsftpd`). Se la regola si trova nel file **`hosts.allow`**, la connessione viene accettata. Se invece si trova in **`hosts.deny`**, la connessione viene rifiutata.

L'esempio successivo è leggermente più complesso, accettando due opzioni:

```
sshd : .example.com  \ : spawn /bin/echo `/bin/date` access denied>>/var/log/sshd.log \ :
 deny
```

Notare la presenza del carattere backslash (\) davanti ad ogni opzione. L'uso del backslash evita che una regola fallisca, a causa della sua lunghezza per un errore sintattico.

Questa regola stabilisce di intercettare ogni host del dominio `example.com` che tenti una connessione con il demone SSH (`sshd`), nel qual caso, il comando **`echo`** trascrive ora e data del tentativo nel file di log specificato e la connessione viene impedita. Poichè si usa la direttiva opzionale **`deny`**, questa regola vieta l'acceso anche se si trova nel file **`hosts.allow`**. Per un analisi più dettagliata sulle opzioni disponibili, vedere la *Sezione 2.5.2.2, «Campi Opzioni»*.

2.5.2.1.1. Wildcards

I termini riservati o wildcard, permettono ai TCP Wrapper di intercettare più facilmente gruppi di demoni o host. Essi sono impiegati frequentemente nel campo della lista dei client di una regola.

I termini riservati sono:

- **`ALL`** — Intercetta tutto. Può essere usato sia nelle lista dei demoni sia in quella dei client.
- **`LOCAL`** — Intercetta tutti gli host il cui hostname non contiene un punto (.), come localhost.
- **`KNOWN`** — Intercetta tutti gli host di cui si conosce l'hostname e l'indirizzo o l'utente.
- **`UNKNOWN`** — Intercetta tutti gli host di cui si non conosce l'hostname o l'indirizzo o l'utente.
- **`PARANOID`** — Intercetta tutti gli host il cui hostname non corrisponde all'indirizzo host.

Importante

I termini **`KNOWN`**, **`UNKNOWN`** e **`PARANOID`** dovrebbero essere impiegati con attenzione, poichè il loro corretto funzionamento si basa su server DNS. Ogni fallimento nella risoluzione di un nome impedisce ad utenti legittimati di ottenere l'accesso al servizio richiesto.

2.5.2.1.2. Pattern

I pattern possono essere usati nel campo della lista dei client, per specificare gruppi di client.

Di seguito si riporta una elenco di pattern comuni:

- *Hostname che iniziano con un punto (.)* — Ponendo un punto davanti ad un hostname, si intercettano tutti gli host che condividono le stesse componenti del nome. Il seguente esempio si applica ad ogni host del dominio `example.com`:

```
ALL : .example.com
```

- *Indirizzo IP con un punto (.) finale* — Inserendo un punto finale ad un indirizzo IP si intercettano tutti gli host che condividono lo stesso gruppo numerico iniziale dell'indirizzo IP. Il seguente esempio si applica a tutti gli host della rete `192.168.x.x`:

```
ALL : 192.168.
```

- *Coppia indirizzo-IP/netmask* — Le netmask possono essere usate come pattern per controllare gli accessi di un particolare gruppo di indirizzi IP. Per esempio la riga seguente si applica ad ogni host che rientri nel range di indirizzi `192.168.0.0` - `192.168.1.255`:

```
ALL : 192.168.0.0/255.255.254.0
```

Importante

Se si opera nello spazio di indirizzamento IPv4, non si può usare la coppia indirizzo/lunghezza-del-prefisso (*prefixlen*) (in notazione CIDR). Soltanto le regole IPv6 possono avvalersi di questo formato.

- *Coppia [IPv6 address]/prefixlen* — Le coppie [net]/prefixlen possono essere usate come pattern per controllare l'accesso di un particolare gruppo di indirizzi IPv6. Il seguente esempio si applica ad ogni host, con un indirizzo compreso tra `3ffe:505:2:1::` e `3ffe:505:2:1:ffff:ffff:ffff:ffff`:

```
ALL : [3ffe:505:2:1::]/64
```

- *L'asterisco (*)* — I caratteri asterisco possono essere usati per intercettare interi gruppi di hostname o indirizzi IP, purchè non siano mescolati in una lista di client, contenenti altri tipi di pattern. Il seguente esempio si applica ad ogni host del dominio `example.com`:

```
ALL : *.example.com
```

- *Lo slash (/)* — Se una lista di client inizia con uno slash, esso viene trattato come un nome di file. Ciò è molto utile quando occorre specificare un gran numero di host. Il seguente esempio riguarda il file **`/etc/telnet.hosts`**:

```
in.telnetd : /etc/telnet.hosts
```

Esistono anche altri pattern, di uso meno frequente. Per maggiori informazioni, fare riferimento alle pagine di man 5, relative a **`hosts_access`**.

Attenzione

Prestare molta attenzione a quando si usano hostname e nomi di dominio. Gli attaccanti possono usare una varietà di trucchi per ingannare il server DNS. Inoltre, l'errato funzionamento del DNS impedisce anche agli utenti autorizzati di usare i servizi di rete. Si raccomanda quindi di usare, quando possibile, indirizzi IP.

2.5.2.1.3. Portmap e TCP Wrapper

L'implementazione di TCP Wrapper per **`portmap`** non supporta l'host look-up (risoluzione di un IP da un hostname), perciò **`portmap`** non può usare l'hostname per identificare l'host. Di conseguenza, le regole di controllo di portmap nei file **`hosts.allow`** o **`hosts.deny`** devono usare indirizzi IP o il termine riservato **`ALL`**, per specificare gli host.

Inoltre, le modifiche alle regole di controllo in **`portmap`** non hanno effetto immediato, ma occorre riavviare il servizio **`portmap`** perchè le modifiche abbiano effetto.

Servizi ampiamente usati come NIS ed NFS, dipendono da **`portmap`** per poter funzionare: si tenga conto di queste limitazioni.

2.5.2.1.4. Operatori

Attualmente, le regole di controllo accettano un solo operatore, **`EXCEPT`**. Può essere usato sia nell'elenco dei demoni di una regola sia in quello dei client.

L'operatore **`EXCEPT`** permette di includere nell'ambito di una regola specifiche eccezioni, estendendo/restringendo il suo campo d'azione.

Nel seguente esempio, gli host del dominio `example.com` escluso `cracker.example.com`, possono connettersi a tutti i servizi:

```
ALL: .example.com EXCEPT cracker.example.com
```

In quest'altro esempio, estratto da un file **`hosts.allow`**, i client della rete `192.168.0.`*`x`* possono usare tutti i servizi, escluso FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

Nota

Per questioni pratiche, si consiglia un uso moderato dell'operatore **`EXCEPT`**, onde evitare agli amministratori (colleghi) di ricercare *anche* gli host esclusi dall'operatore **`EXCEPT`**, tra quelli autorizzati e quelli non autorizzati.

2.5.2.2. Campi Opzioni

L'implementazione in Fedora dei TCP Wrapper, oltre alle regole di base per specificare permessi o divieti d'accesso, supporta estensioni al linguaggio di controllo usando *option fields*. Usando questi campi, si può modificare il livello dei messaggi di log, consolidare il controllo ed avviare comandi di shell.

2.5.2.2.1. Logging

I campi opzione permettono di modificare il comportamento e il livello di priorità dei messaggi di log di una regola, usando la direttiva **`severity`**.

Nel seguente esempio, i messaggi di log per le connessioni dal dominio `example.com` e dirette verso il demone SSH, sono registrate nella facility predefinita **`authpriv`** (non essendo specificato un valore per la facility), di **`syslog`** con priorità **`emerg`**:

```
sshd : .example.com : severity emerg
```

E' anche possibile specificare una facility usando l'opzione **`severity`**. Il seguente esempio registra i messaggi di log di ogni connessione SSH dal dominio `example.com` nella facility **`local0`** con priorità **`alert`**:

```
sshd : .example.com : severity local0.alert
```

Nota

Perchè l'esempio funzioni, occorre che il demone `syslogd` sia configurato per registare i messaggi di log nella facility **`local0`**. Per maggiori informazioni sulla configurazione di messaggi di log non predefiniti, vedere le pagine di man su **`syslog.conf`**.

2.5.2.2.2. Controllo d'Accesso

I campi opzione con la direttiva **`allow`** o **`deny`** posta alla fine di una regola, consentono esplicitamente di autorizzare o vietare host.

Per esempio le seguenti due regole, autorizzano le connessioni SSH da `client-1.example.com`, mentre negano le identiche connessioni da `client-2.example.com`:

```
sshd : client-1.example.com : allow
sshd : client-2.example.com : deny
```

Quindi partendo da una regola base, il campo opzione consente di consolidare tutte le regole d'accesso in un singolo file: nel file **hosts.allow** o nel **hosts.deny**. Per alcuni amministratori tale metodo è una maniera semplice di organizzare le regole d'accesso.

2.5.2.2.3. Comandi di shell

I campi opzione, attraverso le seguenti due direttive, permettono di avviare comandi di shell:

- **spawn** — Avvia un comando di shell come un processo figlio. Questa direttiva può essere usata, per esempio, con il comando **/usr/sbin/safe_finger** per ottenere maggiori informazioni sul client o per creare speciali file di log, usando il comando **echo**.

  Nel seguente esempio, si registrano in un speciale file di log, i client del dominio `example.com` che tentano di accedere al servizio Telnet:

  ```
  in.telnetd : .example.com \
   : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
   : allow
  ```

- **twist** — Sostituisce il servizio richiesto con il comando specificato. Questa direttiva è spesso usata per impostare trappole per intrusori (anche dette "honey pots"). Può essere usata anche per inviare messaggi ai client. La direttiva **twist** deve essere inserita alla fine della regola.

  Nel seguente esempio, i client del dominio `example.com` che tentano di accedere al servizio FTP sono avvisati con un messaggio, usando il comando **echo**:

  ```
  vsftpd : .example.com \
   : twist /bin/echo "421 This domain has been black-listed. Access denied!"
  ```

Per maggiori informazioni sulle opzioni dei comandi di shell, fare riferimento alle pagine di man relative a **hosts_options**.

2.5.2.2.4. Espansioni

Le espansioni quando usate insieme alle direttive **spawn** e **twist**, forniscono informazioni su client, server e processi coinvolti.

Di seguito si riporta un elenco di espansioni supportate:

- **%a** — Restituisce l'indirizzo IP del client
- **%A** — Restituisce l'indirizzo IP del server
- **%c** — Restituisce varie informazioni sul client, come username e hostname, o username e indirizzo IP
- **%d** — Restituisce il nome del processo
- **%h** — Restituisce l'hostname (o l'IP, se l'hostname non è disponibile), del client
- **%H** — Restituisce l'hostname (o l'IP, se l'hostname non è disponibile), del server
- **%n** — Restituisce l'hostname del client. Se non è disponibile, viene restituito **unknown**. Se l'hostname e l'indirizzo non coincidono, viene restituito **paranoid**.
- **%N** — Restituisce l'hostname del server. Se non è disponibile, viene restituito **unknown**. Se l'hostname e l'indirizzo non coincidono, viene restituito **paranoid**.

- **%p** — Restituisce l'ID del processo.
- **%s** — Restituisce varie informazioni sul server, come il processo demone e l'hostname o l'IP del server.
- **%u** — Restituisce lo username del client. Se non è disponibile, viene restituito **unknown**.

Nel seguente esempio, si usa una espansione con il comando **spawn**, per identificare l'host del client che viene registrato in un file di log speciale.

Ogni tentativo di connessione al servizio SSH (sshd), da un host del dominio example.com, lancia il comando **echo** che registra il tentativo, con l'hostname del client (usando l'espansione **%h**), in un file speciale:

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
 : deny
```

In modo analogo, le espansioni possono essere usate per personalizzare i messaggi inviati al client. Nel seguente esempio, i client che tentano di accedere ai servizi FTP dal dominio example.com, vengono informati di essere stati bloccati (banned) dal server:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Per una completa spiegazione delle espansioni, come pure sulle ulteriori opzioni di controllo d'accesso, fare riferimento alle pagine di man 5, relative a **hosts_access** (**man 5 hosts_access**) ed alle pagine di man su **hosts_options**.

Per maggiori informazioni sui TCP Wrapper, fare riferimento alla *Sezione 2.5.5, «Ulteriori risorse»*.

2.5.3. xinetd

Il demone xinetd è un *super servizio* TCP-wrapped, che controlla gli accessi in un sotto-gruppo di servizi di uso comune come FTP, IMAP e Telnet. Fornisce anche, per servizi specifici, opzioni di configurazione per controllo d'accesso, messaggi di log, binding, redirection e per l'utilizzo delle risorse.

Quando un client tenta di connettersi ad un servizio di rete controllato da xinetd, il super servizio prende la richiesta e controlla le regole imposte dal TCP Wrapper.

Se l'accesso è consentito, successivamente xinetd controlla che la connessione sia permessa dalle proprie regole d'accesso. Inoltre controlla se il servizio possa allocare più risorse di quelle consentite e se infranga una qualche regola.

Se sono soddisfatte tutte queste condizioni (ossia, è consentito l'accesso; il servizio non supera le risorse allocabili; ed il servizio di rete non infrange nessuna regola), allora xinetd avvia una istanza del servizio di rete, passando il controllo della connessione al servizio di rete. Una volta stabilita la connessione, xinetd termina la propria partecipazione alla comunicazione tra client e server.

2.5.4. File di configurazione di xinetd

I file di configurazione di xinetd sono i seguenti:

- **/etc/xinetd.conf** — Il file di configurazione globale di xinetd.
- **/etc/xinetd.d/** — La directory con tutti i file di servizio specifici.

2.5.4.1. Il file /etc/xinetd.conf

Il file **`/etc/xinetd.conf`** contiene le impostazioni di configurazione generale dei serivizi controllati da `xinetd`. Esso viene letto al primo avvio di `xinetd`, perciò ogni variazione alla configurazione richiede il riavvio di `xinetd`. Di seguito si riporta un estratto di un file **`/etc/xinetd.conf`**:

```
defaults
{
  instances                = 60
  log_type                 = SYSLOG authpriv
  log_on_success           = HOST PID
  log_on_failure           = HOST
  cps                      = 25 30
}
includedir /etc/xinetd.d
```

Le righe controllano i seguenti aspetti di `xinetd`:

- **`instances`** — Specifica il numero massimo di richieste simultanee processate da `xinetd`
- **`log_type`** — Specifica di usare la facility di log **`authpriv`** che invia i messaggi di log nel file **`/var/log/secure`**. Aggiungendo una direttiva del tipo **`FILE /var/log/xinetdlog`**, `xinetd` crea un file di log specifico di nome **`xinetdlog`** nella directory **`/var/log/`**.
- **`log_on_success`** — Specifica di registrare tutte le connessioni riuscite. Per impostazione, sono registrati l'indirizzo IP dell'host remoto e l'ID di processo del servizio richiesto.
- **`log_on_failure`** — Specifica di registrare le connessioni non riuscite o negate.
- **`cps`** — Specifica di accettare al massimo 25 connessioni al secondo per servizio. Superato il limite, il servizio viene fermato per 30 secondi.
- **`includedir /etc/xinetd.d/`** — Specifica di includere le opzioni dichiarate nei file di configurazione dei servizi, contenuti nella directory **`/etc/xinetd.d/`**. (Vedere la *Sezione 2.5.4.2, «La directory /etc/xinetd.d/»*).

Nota

Spesso, le impostazioni **`log_on_success`** e **`log_on_failure`**, nel file **`/etc/xinetd.conf`**, vengono influenzate dai file di configurazione dei servizi specifici. Quindi, un file di log di un dato servizio può risultare molto più ricco di informazioni di quanto richiesto dalle sole impostazioni di **`/etc/xinetd.conf`**. Per maggiori informazioni, vedere la *Sezione 2.5.4.3.1, «Opzioni di log»*.

2.5.4.2. La directory /etc/xinetd.d/

La directory **`/etc/xinetd.d/`** contiene i file di configurazione di tutti i servizi gestiti da `xinetd`. Analogamente a **`xinetd.conf`**, questa directory è letta al primo avvio di `xinetd`. Ogni modifica ai file di configurazione richiede il riavvio di `xinetd`.

Il formato dei file in **`/etc/xinetd.d/`** usa le stesse convenzioni del file **`/etc/xinetd.conf`**. Il motivo principale che porta ad avere file di configurazione distinti per servizio è di rendere i servizi meno soggetti ad influenze reciproche e di facilitare la loro configurazione.

Per meglio comprendere la struttura interna di questi file, si consideri il file **`/etc/xinetd.d/krb5-telnet`**:

```
service telnet
{
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  disable         = yes
}
```

Le linee controllano vari aspetti del servizio **telnet**:

- **service** — Specifica il nome del servizio, generalmente uno dei servizi presenti nel file **/etc/services**.
- **flags** — Imopsta un attributo sulla connessione. Per esempio l'attributo **REUSE** specifica di riusare il socket per una connessione Telnet.

Nota

L'uso del flag **REUSE** è deprecato. Tutti i servizi ora usano implicitamente il flag **REUSE**.

- **socket_type** — Imposta il tipo di socket, in questo caso **stream**.
- **wait** — Specifica se il servizio è single-thread (**yes**) o multi-thread (**no**).
- **user** — Specifica l'ID utente che ha avviato il processo.
- **server** — Specifica l'eseguibile da avviare.
- **log_on_failure** — Specifica i parametri dei messaggi di log di **log_on_failure**, integrando quelli già definiti in **xinetd.conf**.
- **disable** — Specifica se il servizio è disabilitato (**yes**) o abilitato (**no**).

Per maggiori informazioni sulle opzioni disponibili, consultare le pagine di man relative a **xinetd.conf**.

2.5.4.3. Modificare i file di configurazione di xinetd

I servizi protetti da `xinetd` dispongono di una serie di direttive. Questa sezione illustra quelle maggiormente usate.

2.5.4.3.1. Opzioni di log

Le seguenti opzioni di log sono impiegabili sia in **/etc/xinetd.conf** sia nei file di configurazione della directory **/etc/xinetd.d/** per i particolari servizi.

Le opzioni di logging più comunemente usate sono:

- **ATTEMPT** — Registra un tentativo di connessione fallito (**log_on_failure**).
- **DURATION** — Registra per quanto tempo è stato usato il servizio (**log_on_success**).

- **`EXIT`** — Registra lo stato d'uscita o il segnale di interruzione del servizio (**`log_on_success`**).
- **`HOST`** — Registra l'indirizzo IP dell'host remoto (**`log_on_failure`** e **`log_on_success`**).
- **`PID`** — Registra l'ID del processo server (**`log_on_success`**).
- **`USERID`** — Registra l'utente remoto secondo il metodo definito in RFC 1413 per i servizi stream multi-thread (**`log_on_failure`** e **`log_on_success`**).

Per l'elenco completo delle opzioni di log, fare riferimento alle pagine di man relative a **`xinetd.conf`**.

2.5.4.3.2. Opzioni per il controllo d'accesso

Gli utenti dei servizi di `xinetd` possono scegliere di usare regole d'accesso basate su TCP Wrapper, sui file di configurazione di `xinetd` o su una combinazione di entrambi. Per maggiori informazioni sui file di controllo d'accesso basati su TCP Wrapper, fare riferimento alla *Sezione 2.5.2, «File di configurazione di TCP Wrapper»*.

Questa sezione spiega l'uso di `xinetd` per controllare l'accesso ai servizi.

Nota

Diversamente dai TCP Wrapper, le modifiche al controllo d'accesso hanno effetto solo dopo il riavvio del servizio `xinetd`.

Inoltre, diversamente dai TCP Wrapper, il controllo d'accesso basato su `xinetd`, influenza solo i servizi controllati da `xinetd`.

Il controllo d'accesso di `xinetd` differisce dal metodo usato dai TCP Wrapper. Mentre per i TCP Wrapper le configurazioni di controllo d'accesso si trovano nei due file **`/etc/hosts.allow`** e **`/etc/hosts.deny`**, per `xinetd` le configurazioni si trovano in file distinti, uno per ciascun servizio, nella directory **`/etc/xinetd.d/`**.

`xinetd` supporta le seguenti opzioni d'accesso:

- **`only_from`** — Specifica gli host autorizzati ad usare il servizio.
- **`no_access`** — Specifica gli host non autorizzati ad usare il servizio
- **`access_times`** — Specifica il periodo in cui il servizio è disponibile, secondo il formato HH:MM-HH:MM, dove HH = 00, 01 ... 24.

Le opzioni **`only_from`** e **`no_access`** possono specificare un elenco di indirizzi IP o hostname, o anche specificare una rete. Analogamente ai TCP Wrapper, combinando controlli d'accesso di `xinetd` con opportune configurazioni dei messaggi di log, ripettivamente per bloccare le richieste da host indesiderati e registrare i vari tentativi di accesso, contribuisce a garantire una maggiore sicurezza al sistema.

Per esempio, il seguente file **`/etc/xinetd.d/telnet`** può essere usato per bloccare le connessioni Telnet da una particolare rete e limitare il periodo di connessione agli utenti autorizzati:

```
service telnet
{
  disable         = no
  flags           = REUSE
```

```
   socket_type     = stream
   wait            = no
   user            = root
   server          = /usr/kerberos/sbin/telnetd
   log_on_failure  += USERID
   no_access       = 172.16.45.0/24
   log_on_success  += PID HOST EXIT
   access_times    = 09:45-16:15
}
```

Nell'esempio, quando un client, con indirizzo `172.16.45.2`, tenta di accedere dalla rete `172.16.45.0/24` al servizio Telnet, egli riceve il seguente messaggio:

```
Connection closed by foreign host.
```

Inoltre, i suoi tentativi d'accesso vengono registrati nel file **`/var/log/messages`** come segue:

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address from=172.16.45.2
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285 from=172.16.45.2
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285 duration=0(sec)
```

Quando si usano TCP Wrapper insieme ai controlli d'accesso di `xinetd`, è importante capire il legame tra i due meccanismi di controllo d'accesso.

Di seguito si mostra la sequenza di eventi attivati da `xinetd` quando un client richiede di effettuare una connessione:

1. Il demone `xinetd` analizza le regole d'accesso basate su TCP Wrapper, caricando la libreria **`libwrap.a`**. Se una regola vieta l'accesso, la connessione viene scartata. Se una regola consente l'accesso, il controllo passa a `xinetd`.

2. Il demone `xinetd` controlla le proprie regole d'accesso sia per il servizio di `xinetd` sia per il servizio richiesto. Se esiste una regola di divieto, la connessione viene scartata. Altrimenti, `xinetd` avvia una istanza del servizio e passa il controllo della connessione al servizio.

Importante

Occorre prestare una certa attenzione ad utilizzare controlli d'accesso di TCP Wrapper in combinazione con i controlli di `xinetd`. Effetti indesiderati possono verificarsi in caso di errate configurazioni.

2.5.4.3.3. Opzioni di Binding e di Redirection

I file di configurazione dei servizi di `xinetd`, supportano il collegamento del servizio con un indirizzo IP e la redirezione verso altri indirizzi IP, hostname o porte.

Il collegamento è controllato con l'opzione **`bind`** nei file di configurazione dei servizi e serve a collegare il servizio ad un indirizzo IP nel sistema. Con tale opzione, solo gli host con richieste dirette all'IP specificato possono accedere al servizio. Si può usare questo metodo per collegare p.e. diversi servizi su differenti schede di rete.

Ciò si rivela particolarmente vantaggioso nei sistemi con schede di rete mulltiple o con indirizzi IP multipli. In tali sistemi, servizi non sicuri come Telnet, possono essere configurati (p.e.) per ricevere connessioni soltanto dalla scheda connessa ad una rete privata e non dalla scheda connessa ad Internet.

L'opzione **`redirect`** accetta un indirizzo IP o hostname seguito da un numero di porta. Tale opzione consente di dirottare ogni richiesta di un servizio verso un host e una porta specifica. Questa caratteristica può essere usata per puntare ad un'altra porta del sistema, per redirezionare la richiesta verso un IP differente sulla stessa macchina, per trasferire la richiesta su un sistema completamente diverso oppure può essere usata combinando alcune di queste possibilità. Un utente che si connette al servizio, in maniera trasparente, viene trasferito su un altro sistema senza alcuna interruzione.

Il demone `xinetd` effettua questa redirezione generando un processo, per il trasferimento dei dati tra i due sistemi, che dura quanto la connessione tra la macchina client richiedente e l'host del servizio.

I vantaggi forniti dalle opzioni **`bind`** e **`redirect`**, diventano ancora più evidenti quando le opzioni vengono impiegate insieme. Collegando un servizio ad un particolare indirizzo IP di un sistema e poi reindirizzando le richieste verso una seconda macchina che solo la prima può vedere, un sistema interno può essere usato per fornire servizi ad una rete completamente diversa. Alternativamente, queste opzioni possono essere usate per limitare l'esposizione di un servizio su una macchina multi-homed, ad un indirizzo IP noto, oppure per reindirizzare le richieste verso un'altra macchina, appositamente configurata.

Per esempio, si consideri un sistema usato come firewall con questa impostazione per Telnet:

```
service telnet
{
  socket_type  = stream
  wait   = no
  server   = /usr/kerberos/sbin/telnetd
  log_on_success  += DURATION USERID
  log_on_failure  += USERID
  bind                  = 123.123.123.123
  redirect              = 10.0.1.13 23
}
```

Le opzioni **`bind`** e **`redirect`** assicurano che il servizio Telnet sulla macchina sia collegato all'indirizzo IP esterno `123.123.123.123`, verso Internet. Inoltre, ogni richiesta di servizio Telnet inviata all'indirizzo `123.123.123.123`, viene rediretta, attraverso una seconda scheda di rete, all'indirizzo IP interno `10.0.1.13` a cui possono accedere soltanto il firewall e i sistemi interni. Il firewall quindi gestisce la comunicazione tra i due sistemi, e cosa importante, in maniera trasparente al sistema richiedente che ritiene di comunicare con `123.123.123.123`, quando in realtà è connesso con una macchina differente.

Questa caratteristica è particolarmente utile per quegli utenti con connessioni a banda larga e con un solo indirizzo IP. Quando si usa NAT (Network Address Translation), i sistemi dietro al gateway che usano solo indirizzi IP interni, non sono disponibili dall'esterno. Comunque, se certi servizi controllati da `xinetd` vengono configurati con le opzioni **`bind`** e **`redirect`**, il gateway può agire da proxy tra i sistemi esterni ed una macchina interna configurata per fornire un servizio. Inoltre, le varie opzioni di log e di controllo d'accesso di `xinetd` sono disponibili per fornire ulteriore protezione al sistema.

2.5.4.3.4. Opzioni per gestire le risorse

Il demone `xinetd` può creare una protezione di base contro attacchi tipo DoS (Denial of Service). Di seguito si riporta un elenco di direttive che aiutano a limitare i rischi di tali attacchi:

- **`per_source`** — Definisce il numero massimo di istanze di un servizio, per indirizzo IP ricevente. Accetta solo interi e si può usare sia nel file **`xinetd.conf`** sia nei file di configurazione dei servizi, nella cartella **`xinetd.d/`**.
- **`cps`** — Definisce il numero massimo di connessioni per secondo. La direttiva prende due argomenti di tipo intero, separati da spazio. Il primo argomento rappresenta il numero massimo di connessioni al secondo, per un servizio. L'altro argomento è il numero di secondi di interruzione di `xinetd`,

prima di riabilitare il servizio. Accetta solo interi e si può usare sia nel file **xinetd.conf** sia nei file di configurazione dei servizi, nella cartella **xinetd.d/**.

- **max_load** — Definisce il carico medio da assegnare alla CPU per un servizio. Accetta come argomento un numero decimale (in virgola mobile).

  Il carico medio è una misura (grossolana) del numero dei processi attivi in un dato momento. Per maggiori informazioni sul carico medio di una CPU, vedere le pagine man relative ai comandi **uptime**, **who** e **procinfo**.

Esistono anche altre opzioni per la gestione delle risorse. Per maggiori informazioni, fare riferimento alle pagine di man relative a **xinetd.conf**.

2.5.5. Ulteriori risorse

Maggiori informazioni sui TCP Wrapper e `xinetd` sono disponibili nella documentazione installata nel sistema e su Internet.

2.5.5.1. Documentazione su TCP Wrapper installata

La documentazione installata nel proprio sistema, è un buon punto da cui ottenere informazioni su ulteriori opzioni di configurazione per TCP Wrapper, `xinetd` e controllo d'accesso.

- **/usr/share/doc/tcp_wrappers-<*version*>/** — Questa directory contiene un file **README** che spiega il funzionamento dei TCP Wrapper e i vari rischi relativi alla manomissione (spoofing) degli hostname e degli indirizzi IP degli host.

- **/usr/share/doc/xinetd-<*version*>/** — Questa directory contiene un file **README** che spiega vari aspetti del controllo d'accesso e un file **sample.conf** con vari spunti per modificare i file di configurazione dei servizi, nella directory **/etc/xinetd.d/**.

- Pagine di man su TCP Wrapper e `xinetd` — Esistono un certo numero di pagine di man, dedicate alle varie applicazioni e ai vari file di configurazione rigurdanti TCP Wrapper e `xinetd`. Di seguito si riportano le più importanti:

  Applicazioni server
  - **man xinetd** — Le pagine di man su `xinetd`.

  File di configurazione
  - **man 5 hosts_access** — Le pagine di man sui file di controllo d'accesso di TCP Wrapper.
  - **man hosts_options** — Le pagine di man su option field di TCP Wrapper.
  - **man xinetd.conf** — Le pagine man con l'elenco delle opzioni di configurazione di `xinetd`.

2.5.5.2. Utili siti su TCP Wrapper

- *xinetd*[9] — La home page del progetto, con esempi di file di configurazione, un elenco completo di caratteristiche ed una FAQ informativa.

- *An-Unofficial-Xinetd-Tutorial*[10] — Un tutorial che discute diverse modalità per ottimizzare i file di configurazione di `xinetd` predefiniti, per specifici obbiettivi di sicurezza.

[9] http://www.xinetd.org
[10] http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial

2.5.5.3. Libri

- *Hacking Linux Exposed*, by Brian Hatch, James Lee, and George Kurtz (Osbourne/McGraw-Hill) — E' una eccellente risorsa sulla sicurezza con informazioni su TCP Wrapper e `xinetd`.

2.6. Kerberos

In un sistema di rete, le operazioni necessarie per garantire un livello di sicurezza e di integrità accettabile possono risultare piuttosto impegnative. Anche solo un'analisi per sapere quali servizi siano in esecuzione e in che modo siano utilizzati, può richiedere gli sforzi di alcuni amministratori.

Inoltre, l'autenticazione degli utenti ai servizi di rete può essere rischiosa quando il metodo usato dal protocollo è intrinsecamente insicuro, come nel caso dei protocolli Telnet e FTP che inviano le password in rete senza cifratura.

Kerberos è la maniera di soddisfare il bisogno di autenticazione dei protocolli che usano metodi spesso insicuri, contribuendo così ad aumentare la sicurezza globale della rete.

2.6.1. Cos'è Kerberos?

Kerberos è un protocollo di autenticazione di rete creato dal MIT e che utilizza un sistema di crittografia a chiave simmetrica[11], senza richiedere alcun trasferimento di password.

Di conseguenza, quando gli utenti si autenticano ai servizi che usano Kerberos, viene di fatto impedito ogni possibilità di intercettazione delle password da parte di attaccanti.

2.6.1.1. Vantaggi di Kerberos

I principali servizi di rete usano schemi di autenticazione basati su password, in cui generalmente all'utente viene richiesto di farsi riconoscere con un nome utente e una password. Sfortunatamente, la trasmissione di queste informazioni di autenticazione, per molti servizi avviene in chiaro. Quindi perchè un tale schema sia sicuro, occorre che la rete sia inaccessibile dall'esterno e che tutti gli utenti ed i computer interni siano fidati.

Ma anche nel caso di una rete interna fidata, nel momento in cui viene connessa ad Internet essa non può più considerarsi sicura: un attaccante che riesca ad accedere alla rete, potrebbe usare un semplice analizzatore di pacchetti o packet sniffer, per intercettare nome utente e password, compromettendo gli account utenti e l'integrità della intera rete.

Il principale obbiettivo progettuale di Kerberos è eliminare la trasmissione in chiaro di password; quindi se correttamente configurato, Kerberos effettivamente elimina la minaccia dei packet sniffer.

2.6.1.2. Svantaggi di Kerberos

Anche se Kerberos aiuta a rimuovere comnuni e gravi minacce alla sicurezza, la sua implementazione, per una varietà di ragioni, può risultare complessa:

- Migrare le password utenti da un database di password UNIX (standard), come **`/etc/passwd`** o **`/etc/shadow`** in un database di password Kerberos, può essere un'operazione tediosa, perchè al momento non esiste un meccanismo automatizzato. Fare riferimento alla Question 2.23 della Kerberos FAQ, al seguente link:

  http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html[12]

---

[11] Un sistema in cui sia il client sia il server condividono una chiave comune usata per cifrare/decifrare la comunicazione.
[12] http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html

- Kerberos presenta solo una parziale compatibilità con il sistema PAM (Pluggable Authentication Modules), usato nei principali server Fedora. Per maggiori informazioni, vedere la *Sezione 2.6.4, «Kerberos e PAM»*.

- Kerberos assume che ogni utente sia fidato, in un ambiente in cui generalmente l'utente e la rete non lo sono. Il suo obbiettivo principale è impedire il trasferimento in chiaro di password. Se un utente qualunque non autorizzato, riesce ad accedere ad uno degli host che distribuisce ticket, usati per l'autenticazione — denominato *KDC* (*Key Distribution Center*) — l'intero sistema di autenticazione Kerberos viene messo a rischio.

- Se si vuole che un'applicazione usi Kerberos, il codice sorgente dell'applicazione deve essere opportunamente modificato in modo da poter chiamare le librerie di Kerberos. Le applicazioni così adattate sono dette *Kerberos-aware* o *kerberizzate*. Per alcune applicazioni, ciò può essere problematico per motivi progettuali e dimensionali. Per altre applicazioni incompatibili, le modifiche devono essere fatte tenendo conto delle modalità di comunicazione tra server e client. Di nuovo, ciò potrebbe richiedere notevoli modifiche al codice originario. Le applicazioni closed-source che non supportano Kerberos per impostazione, sono spesso quelle più problematiche.

- Kerberos è una soluzione determinante/decisiva. Se usato in una rete, ogni password trasferita in chiaro ad un servizio non *kerberizzato*, diventa un rischio per la sicurezza. In tal caso, la rete non trae alcun vantaggio dall'uso di Kerberos. Quindi per rendere sicura una rete con Kerberos, *tutte* le applicazioni client/server che trasmettono password in chiaro, devono essere *kerberizzate*.

2.6.2. Terminologia Kerberos

Kerberos ha la propria terminologia per specificare i vari aspetti del servizio. Per capire il funzionamento di Kerberos, è importante conoscere il significato dei seguenti termini.

Authentication Server (AS)
: Un server di distribuzione di ticket che vengono rilasciati al client, per accedere ad un determinato servizio. Un AS risponde alle richieste dei client che non hanno o non hanno trasmesso le credenziali con una richiesta. Di solito è usato per accedere al server TGS (Ticket Granting Server), rilasciando un ticket TGT (Ticket Granting Ticket). Un server AS generalmente si trova sullo stesso host del KDC (Key Distribution Center).

testo cifrato
: Dati crittati o non in chiaro

client
: Una entità sulla rete (un utente, un host o una applicazione), che può ottenere un ticket da Kerberos.

credenziali
: Un insieme di credenziali temporanee, che verificano l'identità di un client per un particolare servizio. Viene anche detto *ticket*.

credential cache o file dei ticket
: Un file contenente le chiavi per cifrare le comunicazioni tra l'utente ed i vari servizi. Kerberos 5 supporta una piattaforma per altri tipi di memorizzazione, come la memoria condivisa, ma i file sono maggiormente supportati.

funzione hash di cifratura
: Una funzione hash usata per trasformare dati. I dati così manipolati, sono più sicuri rispetto ai dati originali, ma restano abbastanza semplici da decifrare da parte di un cracker esperto.

GSS-API
: La GSS-API o Generic Security Service Application Program Interface (pubblicata da The Internet Engineering Task Force in RFC-2743), è un insieme di funzioni che offrono servizi di sicurezza. Questa API, mascherando il meccanismo sottostante, è usata da client e servizi per autenticazione reciproca. Se un servizio come cyrus-IMAP, usa GSS-API, allora esso può autenticarsi via Kerberos.

hash
: Anche detto *valore hash*. E' un valore ottenuto passando una stringa ad una *funzione hash*. Questi valori sono tipicamente usati per essere sicuri che i dati trasmessi non siano stati manomessi.

funzione hash
: Un modo per generare un "fingerprint" o firma su dei dati d'ingresso. Queste funzioni eseguono delle trasformazioni o alterazioni sui dati, producendo un *valore hash*.

chiave
: I dati usati per cifrare o decifrare altri dati. I dati cifrati non possono essere decifrati senza la chiave appropriata o senza una straordinaria fortuna da parte del cracker.

Key Distribution Center (KDC)
: Un servizio che invia ticket Kerberos e generalmente esegue sullo stesso host del TGS (Ticket Granting Server).

keytab (o tabella delle chiavi)
: Un file contenente una lista in chiaro di *principal* e delle loro chiavi. Un server ottiene le chiavi necessarie dal file keytab invece di usare **`kinit`**. Il file keytab predefinto è **`/etc/krb5.keytab`**. Il server d'amministrazione KDC, **`/usr/kerberos/sbin/kadmind`**, è l'unico servizio che usa un altro file (esso usa **`/var/kerberos/krb5kdc/kadm5.keytab`**).

kinit
: Il comando **`kinit`** consente ad un principal già loggato di ottenere e memorizzare il TGT (Ticket Granting Ticket) iniziale. Per maggiori informazioni su **`kinit`**, consultare le pagine di man relative.

principal (o nome del principal)
: Il principal è il nome unico di un utente o servizio, abilitato ad autenticarsi presso Kerberos. Un principal segue la forma di **`root[/instance]@REALM`**. Per un utente tipico, **`root`** coincide con l'ID associato all'account utente. Il termine **`instance`** è opzionale. Se il principal ha un **`instance`**, esso viene separato dal **`root`**, usando un carattere "forward slash" ("/"). Una stringa vuota ("") è considerata un **`instance`** valido (differente dall'instance predefinito, **`NULL`**), tuttavia il suo utilizzo può essere fonte di confusione. Tutti i principal di un realm hanno la propria chiave, derivata da una password se si tratta di utenti o impostata casualmente se si tratta di servizi.

realm
: Una rete che usa Kerberos, composta da uno o più server KDC e un numero potenzialmente grande di client.

servizio
: Un programma accessibile dalla rete.

ticket
: Un insieme di credenziali temporanee che verificano l'identità di un client per un particolare servizio. Viene anche detto credenziali.

Ticket Granting Server (TGS)
: Un server che distribuisce ticket per un servizio, girati agli utenti per accedere al servizio. Generalmente un TGS esegue sullo stesso host che ospita il KDC.

Ticket Granting Ticket (TGT)
: Un ticket speciale che consente al client di ottenere ulteriori ticket senza dover inoltrare le richieste al KDC.

password non cifrata
: Una password in chiaro o leggibile.

2.6.3. Come funziona Kerberos

Kerberos differisce dai tradizionali metodi di autenticazione basati su nome-utente/password. Infatti, invece di autenticare l'utente per ogni servizio, Kerberos usa un sistema di crittografia simmetrica e un terzo fidato (un KDC) per autenticare gli utenti ai vari servizi di rete. Quando un utente si autentica presso il KDC, il KDC restituisce, alla macchina dell'utente, un ticket specifico valido per la sessione ed ogni servizio kerberizzato cerca il ticket sulla macchina del client, invece di richiedere all'utente di autenticarsi con una password.

Quando l'utente avvia una sessione su una workstation in una rete controllata da Kerberos, il suo principal viene trasmesso al KDC per una richiesta di TGT, da parte dell'Authentication Server. Questa richiesta può venir trasmessa dal programma di log-in o venir trasmessa dal programma **`kinit`**, ad accesso avvenuto.

A questo punto il KDC controlla il principal nel proprio database. Se il principal esiste, il KDC crea un TGT, che viene cifrato con la chiave dell'utente e restituito all'utente.

Poi il programma di log-in o **`kinit`**, decifra il TGT usando la chiave dell'utente, ottenuta dalla password dell'utente. Quindi la chiave dell'utente è usata soltanto sulla macchina del client e *non* viene trasmessa nella rete.

Sul TGT viene imposta una scadenza (usualmente tra dieci e ventiquattro ore), dopodichè viene conservato nella credential cache della macchina del client. La scadenza serve a limitare il periodo a disposizione di un eventuale attaccante, che sia entrato in possesso di un TGT compromesso. Una volta ottenuto il TGT, l'utente non deve re-inserire la propria password fino alla scadenza del TGT, a meno che non esca e rientri in una nuova sessione.

Ogni volta che l'utente accede ad un servizio, il client usa il TGT per richiedere al TGS un nuovo ticket per quel determinato servizio. Il ticket è poi usato per autenticare l'utente al servizio.

Avviso

Il sistema Kerberos può essere compromesso se un utente si autentica presso un servizio non kerberizzato, trasmettendo una password in chiaro. L'utilizzo di un servizio non kerberizzato è fortemente scoraggiato. Tali servizi includono Telent ed FTP. L'utilizzo di altri protocolli cifrati, come i servizi sicuri SSH o SSL, comunque sono da preferirsi, sebbene non ideali.

Quanto finora esposto, è soltanto una breve panoramica su come funziona l'autenticazione di Kerberos. Per maggiori informazioni fare riferimento ai link nella *Sezione 2.6.10, «Ulteriori risorse»*.

Nota

Per poter funzionare correttamente, Kerberos necessita dei seguenti servizi di rete:

- Sincronizzazione approssimata del clock tra le macchine di rete.

  Nella rete dovrebbe essere configurato un programma di sincronizzazione del clock, come **`ntpd`**. Per maggiori dettagli su come configurare un server Network Time Protocol, fare riferimento al file **`/usr/share/doc/ntp-<version-number>/index.html`**, dove *`<version-number>`* è la versione del pacchetto **`ntp`** installato.

- DNS (Domain Name Service)

  Assicurarsi che il DNS e gli host sulla rete siano correttamente configurati. Per maggiori informazioni, consultare *Kerberos V5 System Administrator's Guide* nella cartella **`/usr/share/doc/krb5-server-<version-number>`**, dove *`<version-number>`* è la versione del pacchetto **`krb5-server`** installato.

2.6.4. Kerberos e PAM

I servizi kerberizzati, in realtà, non fanno uso di PAM (Pluggable Authentication Modules) — questi servizi by-passano del tutto PAM. Comunque, installando il modulo **`pam_krb5`** (fornito con il pacchetto **`pam_krb5`**), le applicazioni che usano PAM possono far uso di Kerberos per l'autenticazione. Il pacchetto **`pam_krb5`** contiene alcuni file campione da cui è possibile configurare servizi come **`login`** e **`gdm`**, per autenticare gli utenti e per ottenere le credenziali iniziali da password. Se l'accesso ai server di rete avviene sempre tramite servizi kerberizzati o servizi che usano GSS-API, come IMAP, allora la rete può considerarsi ragionevolmente sicura.

Importante

Gli amministratori dovrebbero vietare agli utenti di usare le password di Kerberos, per autenticarsi ai servizi di rete. Molti protocolli usati da questi servizi, non cifrano le password, vanificando i benefici del sistema Kerberos. Per esempio, non si dovrebbe consentire di accedere ai servizi Telnet, con la stessa password usata per autenticarsi presso Kerberos.

2.6.5. Configurare un server Kerberos 5

Quando si imposta Kerberos, installare dapprima il KDC. Se occorre impostare alcuni server slave, installare prima il master.

Per configurare il primo KDC Kerberos, seguire i seguenti passaggi:

1. Prima di configurare Kerberos, assicurarsi che il servizio di sincronizzazione del clock e il DNS, funzionino correttamente su tutti i client e server. Prestare particolare attenzione alla sincronizzazione dell'ora tra il server Kerberos e i suoi client. Se il server ed i client sono sfasati per più di cinque minuti, i client non possono autenticarsi presso il server. Questa sincronizzazione è necessaria in quanto impedisce ad un attaccante, che utilizzi un vecchio ticket, di mascherarsi come un utente fidato.

Si consiglia di impostare un NTP (Network Time Protocol) anche se non si usa Kerberos. In Fedora è incluso nel pacchetto **ntp**. Per i dettagli su come impostare un server Network Time Protocol, fare riferimento al file **/usr/share/doc/ntp-<*version-number*>/index.html**, dove <*version-number*> è la versione del pacchetto **ntp** installato nel proprio sistema, o visitare il sito del progetto *http://www.ntp.org*.

2. Installare i pacchetti **krb5-libs**, **krb5-server** e **krb5-workstation**, sulla macchina che ospiterà il KDC. Questa macchina deve risultare molto sicura — se possibile, si dovrebbe eseguire esclusivamente il servizio KDC.

3. Modificare il nome del realm e le associazioni tra domini e realm, nei file di configurazione **/etc/krb5.conf** e **/var/kerberos/krb5kdc/kdc.conf**. Per creare un semplice realm, sostituire le istanze di *EXAMPLE.COM* e *example.com* con il nome corretto del dominio — tenendo conto che il nome è "case sensitive" — e sostituire *kerberos.example.com* con il nome del server KDC. Per convenzione, tutti i realm sono espressi con lettere maiuscole e tutti gli hostname e i domini in lettere minuscole. Per maggiori dettagli sui formati di questi file di configurazione, fare riferimento alle rispettive pagine di man.

4. Creare il database usando l'utility da terminale, **kdb5_util**:

```
/usr/kerberos/sbin/kdb5_util create -s
```

Il comando **create**, genera il database con le chiavi per il realm Kerberos. Lo switch **-s**, invece, crea un file *stash* in cui è salvata la chiave del server master. Se il file *stash* non viene creato, il server Kerberos (**krb5kdc**) richiede all'utente di inserire la password per il server master (usata per rigenerare la chiave), ad ogni suo avvio.

5. Modificare il file **/var/kerberos/krb5kdc/kadm5.acl**. Questo file, usato dal comando **kadmind**, determina i principal che hanno accesso amministrativo, con i relativi livelli, al database di Kerberos. Generalmente basta una semplice riga:

```
*/admin@EXAMPLE.COM  *
```

Gli utenti, generalmente, sono rappresentati nel database da un unico principal (con instanza *NULL*, o vuota come *joe@EXAMPLE.COM*). Con questa configurazione, gli utenti con un secondo principal con instanza *admin* (per esempio, *joe/admin@EXAMPLE.COM*) possono avere pieno controllo sul database Kerberos del realm.

Dopo aver avviato il server, con il comando **kadmind**, ogni utente può accedere ai suoi servizi eseguendo il comando **kadmin** su un client o su un server del realm. Comunque, solo gli utenti elencati nel file **kadm5.acl**, possono modificare il contenuto del database, ad eccezione delle password.

Nota

L'utility **`kadmin`** comunica con il server **`kadmind`**, ed usa Kerberos per l'autenticazione. Poichè, occorre il primo principal per effettuare una connessione con il server da amministrare, creare il principal con il comando **`kadmin.local`**, specificatamente progettato per essere impiegato sullo stesso host del KDC e che non usa Kerberos per autenticazione.

Per creare il primo principal, nel KDC, digitare il comando **`kadmin.local`**:

```
/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"
```

6. Avviare Kerberos usando i seguenti comandi:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
/sbin/service krb524 start
```

7. Aggiungere i principal degli utenti, usando il comando **`addprinc`** (dall'interfaccia di **`kadmin`**). I comandi **`kadmin`** e **`kadmin.local`**, sono comandi da terminale che si interfaccano con il KDC. Una volta avviato il programma **`kadmin`**, sono disponibili molti altri comandi simili ad **`addprinc`**. Per maggiori informazioni su **`kadmin`**, fare riferimento alla relative pagine di man.

8. Verificare che il KDC emetta ticket. Per prima cosa, lanciare **`kinit`** per ottenere un ticket e conservarlo in un credential cache. Poi, usare il comando **`klist`** per visualizzare la lista delle credenziali in cache, e **`kdestroy`** per rimuovere la lista e la credential cache.

Nota

Per impostazione, **`kinit`** tenta l'autenticazione usando lo stesso nome-utente dell'account di sistema (non del server Kerberos). Se il nome-utente non corrisponde ad un principal del database di Kerberos, **`kinit`** segnala un messaggio d'errore. Per ovviare a questo problema, aggiungere a **`kinit`** come argomento, il nome esatto del principal (**`kinit <principal>`**).

Una volta completati questi passaggi, il server Kerberos dovrebbe essere attivo e funzionante.

2.6.6. Configurare un client Kerberos 5

Impostare un client Kerberos 5 è meno complicato rispetto all'impostazione di un server. Come minimo, installare i pacchetti del client e fornire ogni client di un file di configurazione **`krb5.conf`**, valido. Sebbene **`ssh`** e **`slogin`** siano i metodi migliori per accedere da remoto ai client, nel caso esistessere ancora versioni kerberizzate di **`rsh`** ed **`rlogin`**, il loro utilizzo richiederebbe di apportare ulteriori modifiche ai file di configurazione.

1. Assicurarsi che il servizio di sincronizzazione del clock, tra il client Kerberos ed il KDC, funzioni correttamente. (Vedere la *Sezione 2.6.5, «Configurare un server Kerberos 5»*.) Inoltre prima di ogni configurazione, verificare che funzioni il DNS sul client Kerberos.

2. Installare i pacchetti **krb5-libs** e **krb5-workstation** su tutte le macchine client. Fornire ogni macchina di un valido file **/etc/krb5.conf** (normalmente si può usare lo stesso file **krb5.conf** del KDC).

3. Prima che una workstation del realm possa usare Kerberos, per autenticare gli utenti ai servizi **ssh** o a versioni kerberizzate di **rsh** o **rlogin**, essa deve possedere il principal del proprio host, nel database di Kerberos. I server **sshd**, **kshd** e **klogind** necessitano tutti di accedere alle chiavi del principal del servizio *host*. Inoltre, per usare i servizi **rsh** ed **rlogin** kerberizzati, la workstation deve avere installato il pacchetto **xinetd**.

   Usando **kadmin**, aggiungere sul KDC, un principal host per la workstation. In questo caso, l'istanza è l'hostname della workstation. Passare l'opzione **-randkey** insieme al comando **addprinc**, per creare il principal ed assegnarli una chiave casuale:

   ```
   addprinc -randkey host/blah.example.com
   ```

   Una volta creato il principal, le chiavi possono essere estratte, eseguendo il comando **kadmin** *sulla workstation stessa*, seguito dal comando **ktadd**:

   ```
   ktadd -k /etc/krb5.keytab host/blah.example.com
   ```

4. Per usare altri servizi kerberizzati, occorre dapprima avviarli. Di seguito si riporta una lista di alcuni comuni servizi kerberizzati e le istruzioni per abilitarli:

   - **ssh** — OpenSSH usa GSS-API per autenticare gli utenti ai servizi, se client e server sono entrambi configurati con l'opzione **GSSAPIAuthentication** abilitata. Se il client è configurato anche con l'opzione **GSSAPIDelegateCredentials** abilitata, le credenziali utente vengono rese disponibili al sistema remoto.

   - **rsh** e **rlogin** — Per usare le versioni kerberizzate di **rsh** ed **rlogin**, abilitare **klogin**, **eklogin** e **kshell**.

   - **Telnet** — Per usare la versione kerberizzata di Telnet, abilitare **krb5-telnet**.

   - **FTP** — Per fornire accesso FTP, creare ed estrarre una chiave per il principal, impostando il root per il principal su **ftp**. Assicurarsi di impostare l'instance con l'hostname completo del server FTP e poi abilitare **gssftp**.

   - **IMAP** — Per usare un server IMAP kerberizzato v.5, occorre installare i pacchetti **cyrus-imap** e **cyrus-sasl-gssapi**. Quest'ultimo contiene i componenti Cyrus SASL che supportano l'autenticazione tramite GSS-API. Cyrus IMAP dovrebbe funzionare correttamente con Kerberos se l'utente **cyrus** è in grado di trovare la chiave appropriata nel file **/etc/krb5.keytab**, ed il root per il principal è impostato su **imap** (creato con **kadmin**).

     Un'alternativa a **cyrus-imap** è data dal pacchetto **dovecot**, incluso anche in Fedora. Questo pacchetto contiene un server IMAP, ma per il momento senza alcun supporto per GSS-API e Kerberos.

   - **CVS** — Per usare un server CVS kerberizzato, **gserver** usa un principal con root impostato su **cvs**; il resto è identico a **pserver** di CVS.

2.6.7. Associazione tra Dominio e Realm

Quando un client tenta di accedere ad un servizio di rete, esso conosce il nome del servizio (*host*) ed il nome del server (*foo.example.com*), ma poichè nella rete può esserci più di un realm, il client deve innanzittutto individuare il nome del realm in cui si trova il servizio.

Per impostazione, il nome del realm coincide con il nome, in lettere maiuscole, del dominio DNS del server.

```
foo.example.org → EXAMPLE.ORG
 foo.example.com → EXAMPLE.COM
 foo.hq.example.com → HQ.EXAMPLE.COM
```

In alcune configurazioni, ciò è sufficiente, ma in altre, il nome del realm derivato coincide con il nome di un realm inesistente. In queste situazioni, l'associazione tra il nome del dominio del server con il nome del suo realm, deve essere specificato nella sezione *domain_realm* del file **krb5.conf**, nel sistema del client. Per esempio:

```
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

La configurazione precedente specifica due associazioni. La prima specifica che ogni sistema nel dominio "example.com" appartiene al realm *EXAMPLE.COM*. La seconda specifica che un sistema con il nome coincidente con "example.com" si trova nello stesso realm. (La distinzione tra un dominio e uno specifico host, è contrassegnata dalla presenza o assenza di un "." iniziale.) L'associazione può essere salvata anche direttamente nel server DNS.

2.6.8. Impostare KDC secondari

Per diverse ragioni, si potrebbe decidere di eseguire più KDC in un dato realm. In questo scenario, un KDC (il *master KDC*) conserva una copia modificabile del database del realm ed esegue **kadmind** (in qualità di *admin server* del realm), ed uno o più KDC (*slave KDC*) conservano copie locali in sola lettura del database, ed eseguono **kpropd**.

La procedura di propagazione master-slave assegna al master KDC il compito di replicare il suo database in un file temporaneo, per poi trasmetterlo a ciascuno dei suoi slave, i quali aggiornano in tal modo il contenuto della loro copia in sola lettura, ricevuta in precedenza, con il contenuto modificabile del master.

Prima di procedere con l'impostazione di uno slave KDC, assicurarsi di copiare su ogni slave KDC i file **krb5.conf** e **kdc.conf** del master KDC.

Avviare **kadmin.local** da una shell di root, sul master KDC, ed usare il comando **add_principal** per creare una nuova istanza del servizio *host* sul master KDC, e poi usare il comando **ktadd** per impostare simultaneamente una chiave casuale per il servizio e salvare la chiave nel file keytab predefinito, sul master. Questa chiave è usata dal comando **kprop** per autenticazioni presso i server slave. Questa operazione va effettuata soltanto una volta, a prescindere dal numero di slave da installare.

```
# kadmin.local -r EXAMPLE.COM

Authenticating as principal root/admin@EXAMPLE.COM with password.

kadmin: add_principal -randkey host/masterkdc.example.com

Principal "host/host/masterkdc.example.com@EXAMPLE.COM" created.
```

```
kadmin: ktadd host/masterkdc.example.com

Entry for principal host/masterkdc.example.com with kvno 3, encryption type Triple DES cbc
 mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type ArcFour with
 HMAC/md5 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type DES with HMAC/
sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type DES cbc mode with
 RSA-MD5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

Avviare **kadmin** da una shell di root sullo slave KDC, ed usare il comando **add_principal** per creare una nuova istanza del servizio *host* sullo slave KDC, e poi usare il comando **ktadd** per impostare simultaneamente una chiave casuale per il servizio e salvare la chiave nel file keytab predefinito sullo slave. Questa chiave è usata dal servizio **kpropd** per autenticare i client.

```
# kadmin -p jimbo/admin@EXAMPLE.COM -r EXAMPLE.COM

Authenticating as principal jimbo/admin@EXAMPLE.COM with password.

Password for jimbo/admin@EXAMPLE.COM:

kadmin: add_principal -randkey host/slavekdc.example.com

Principal "host/slavekdc.example.com@EXAMPLE.COM" created.

kadmin: ktadd host/slavekdc.example.com@EXAMPLE.COM

Entry for principal host/slavekdc.example.com with kvno 3, encryption type Triple DES cbc
 mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type ArcFour with HMAC/
md5 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type DES with HMAC/sha1
 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type DES cbc mode with
 RSA-MD5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

Con il suo servizio chiavi, lo slave KDC potrebbe autenticare ogni client che vorrebbe connettersi. E, con un nuovo database di realm, non a tutti i client dovrebbe essere permesso di usufruire del servizio **kprop** dello slave. Quindi, per limitare l'accesso, il servizio **kprop** sullo slave KDC, accetta aggiornamenti solo per quei client i cui principal sono elencati nel file **/var/kerberos/krb5kdc/kpropd.acl**. Aggiungere a questo file, il nome del servizio host sul master KDC.

```
 # echo host/masterkdc.example.com@EXAMPLE.COM > /var/kerberos/krb5kdc/kpropd.acl
```

Una volta ricevuta una copia del database, lo slave KDC ha bisogno di conoscere la chiave, usata dal master, per cifrarlo. Se la chiave è conservata in un file *stash* sul master KDC (tipicamente nel file **/var/kerberos/krb5kdc/.k5.REALM**), copiarlo sullo slave KDC usando un metodo sicuro, oppure creare un database fasullo e un identico file stash sullo slave KDC, usando il comando **kdb5_util create -s** (il database fasullo verrà sovrascritto alla prima propagazione) e impiegando la stessa password.

Assicurarsi che il firewall dello slave KDC permetta al master KDC di contattare lo slave sulla porta TCP 754 (*krb5_prop*), ed avviare il servizio **kprop**. Poi, verificare attentamente che il servizio **kadmin** sia *disabilitato*.

A questo punto, effettuare un test manuale di propagazione del database, effettundo un *dump* del database del realm sul KDC master, nel file predefinito **/var/kerberos/krb5kdc/slave_datatrans**, letto dal comando **kprop**, e poi usare lo stesso comando per trasmettere il suo contenuto sullo slave KDC.

```
# /usr/kerberos/sbin/kdb5_util dump /var/kerberos/krb5kdc/slave_datatrans# kprop
slavekdc.example.com
```

Con **kinit**, verificare che un client, il cui file di configurazione **krb5.conf** nella lista dei KDC del realm, contiene soltanto il KDC slave, sia in grado di ricevere le credenziali iniziali dallo slave.

Fatto ciò, creare uno script che effettui un *dump* del database del realm ed esegua il comando **kprop**, trasmettendo regolarmente il database ad ogni slave KDC; infine configurare il servizio **cron** per la periodica esecuzione dello script.

2.6.9. Impostare autenticazioni cross realm

Con autenticazione *cross realm*, si indica la situazione in cui i client (tipicamente utenti), di un realm usano Kerberos per autenticarsi ai servizi appartenenti ad un diverso realm (tipicamente i servizi sono processi server in esecuzione su un particolare sistema).

Nel caso più semplice, se un client di un realm di nome **A.EXAMPLE.COM**, vuole accedere ad un servizio del realm **B.EXAMPLE.COM**, entrambi i realm devono condividere una chiave per un principal di nome **krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM**, ed entrame le chiavi devono possedere lo stesso **kvno** (key version number).

Per fare questo, selezionare una password o passphrase molto robusta, e con il comando kadmin, creare un'istanza per il principal in entrambi i realm.

```
# kadmin -r A.EXAMPLE.COM kadmin: add_principal krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM
Enter password for principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM": Re-enter
password for principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM": Principal
"krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM" created. quit # kadmin -r B.EXAMPLE.COM
kadmin: add_principal krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM Enter password for principal
"krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM": Re-enter password for principal "krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM": Principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM" created. quit
```

Usare il comando **get_principal**, per verificare che entrambe le istanze abbiano identici **kvno**) e stesso tipo di cifratura.

Con il dump del Database, non funziona!

Amministratori attenti alla sicurezza, potrebbero essere tentati di usare l'opzione **-randkey** del comando **add_principal**, per assegnare una chiave casuale invece di usare una password; e poi effettuare un dump della nuova istanza, dal database del primo realm ed importarlo nel secondo. Ciò non funziona, a meno che non siano identiche le chiavi master nei database dei realm, poichè le chiavi contenute in un dump del database sono a loro volta cifrate con la chiave master.

I client nel realm **A.EXAMPLE.COM** possono ora autenticarsi presso i servizi del realm **B.EXAMPLE.COM**. In altri termini, il realm **B.EXAMPLE.COM** si fida del realm **A.EXAMPLE.COM**, o più semplicemente **B.EXAMPLE.COM** si fida di **A.EXAMPLE.COM**.

Ciò consente una conclusione importante: la fiducia cross-realm è per impostazione, unidirezionale. Il KDC del realm **B.EXAMPLE.COM** si fida dei client di **A.EXAMPLE.COM** autenticandoli ai servizi nel realm **B.EXAMPLE.COM**, ma questo fatto non dice nulla se i client nel realm **B.EXAMPLE.COM** siano fidati per autenticarsi ai servizi nel realm **A.EXAMPLE.COM**. Per stabilire la fiducia nell'altra direzione, entrambi i realm dovrebbero condividere una chiave per il servizio **krbtgt/A.EXAMPLE.COM@B.EXAMPLE.COM** (notare l'inversione dei due realm, rispetto all'esempio precedente).

Se le relazioni di fiducia dirette, fossero l'unico metodo disponibile per fornire la fiducia fra realm, le reti contenenti realm multipli sarebbero molto difficili da impostare. Fortunatamente, la fiducia cross-realm è transitiva. Se i client di **A.EXAMPLE.COM** possono autenticarsi ai servizi di **B.EXAMPLE.COM** ed i client di **B.EXAMPLE.COM** possono autenticarsi ai servizi di **C.EXAMPLE.COM**, allora anche i client di **A.EXAMPLE.COM** possono autenticarsi ai servizi di **C.EXAMPLE.COM**, anche senza la fiducia diretta tra **C.EXAMPLE.COM** ed **A.EXAMPLE.COM**. Quindi, in una rete con realm multipli cui occorre dare fiducia reciproca, fare delle buone scelte iniziali sulle relazioni di fiducia da accordare, può contribuire a ridurre le complicazioni di configurazione.

Ora occorre affrontare il problema più comune: il sistema del client deve essere configurato in modo da poter dedurre il realm cui appartiene un servizio, e deve essere in grado di determinare, come ottenere le credenziali per i servizi nel realm.

Innazitutto: il nome del principal, per un servizio offerto da un server in un realm, tipicamente ha la seguente struttura:

```
service/server.example.com@EXAMPLE.COM
```

In questo esempio, *service* generalmente rappresenta il nome del protocollo (valori comuni possono essere *ldap*, *imap*, *cvs* ed *HTTP*), o l'*host*; *server.example.com* è il nome di dominio o FQDN del sistema su cui funziona il servizio, ed **EXAMPLE.COM** è il nome del realm.

Per dedurre il realm a cui appartiene il servizio, i client molto spesso consultano il DNS o la sezione **domain_realm** nel file **/etc/krb5.conf**, associando un hostname (*server.example.com*) o un nome di dominio (*.example.com*) al nome del realm (*EXAMPLE.COM*).

Dopo aver individuato il realm cui appartiene un servizio, per ottenere le credenziali da usare per autenticarsi al servizio, il client deve determinare l'insieme dei realm da contattare e sapere in quale ordine contattarli.

Ciò può avvenire in due modi.

Il metodo predefinito, che non richiede esplicita configurazione, è di assegnare ai realm, i nomi di una gerarchia condivisa. Per esempio, si considerino i seguenti realm di nome **A.EXAMPLE.COM**, **B.EXAMPLE.COM** ed **EXAMPLE.COM**. Quando un client del realm **A.EXAMPLE.COM** tenta di autenticarsi presso un servizio di **B.EXAMPLE.COM**, per impostazione, tenta dapprima di ottenere le credenziali per il realm **EXAMPLE.COM**, e poi usando queste credenziali, di ottenere le credenziali per il realm **B.EXAMPLE.COM**.

Il client, in questo scenario, tratta il nome del realm come un nome di DNS. In altre parole, il client rimuove ripetutamente i componenti dal proprio nome di realm, creando i nomi dei realm che si trovano in "cima" alla gerarchia, finchè non raggiunge un punto che si trova in "cima" al realm del servizio. A questo punto incomincia ad anteporre i componenti del nome del servizio, fino ad ottenere il realm del servizio. Ogni realm coinvolto nel processo è un altro "hop" (o salto).

Per esempio, usando le credenziali in **A.EXAMPLE.COM**, un client vuole autenticarsi ad un servizio in **B.EXAMPLE.COMA.EXAMPLE.COM → EXAMPLE.COM → B.EXAMPLE.COM**

- **A.EXAMPLE.COM** e **EXAMPLE.COM** condividono una chiave per **krbtgt/EXAMPLE.COM@A.EXAMPLE.COM**
- **EXAMPLE.COM** e **B.EXAMPLE.COM** condividono una chiave per **krbtgt/B.EXAMPLE.COM@EXAMPLE.COM**

Un altro esempio: usando le credenziali in **SITE1.SALES.EXAMPLE.COM**, un client vuole autenticarsi ad un servizio in **EVERYWHERE.EXAMPLE.COMSITE1.SALES.EXAMPLE.COM → SALES.EXAMPLE.COM → EXAMPLE.COM → EVERYWHERE.EXAMPLE.COM**

- **SITE1.SALES.EXAMPLE.COM** e **SALES.EXAMPLE.COM** condividono una chiave per **krbtgt/SALES.EXAMPLE.COM@SITE1.SALES.EXAMPLE.COM**
- **SALES.EXAMPLE.COM** e **EXAMPLE.COM** condividono una chiave per **krbtgt/EXAMPLE.COM@SALES.EXAMPLE.COM**
- **EXAMPLE.COM** e **EVERYWHERE.EXAMPLE.COM** condividono una chiave per **krbtgt/EVERYWHERE.EXAMPLE.COM@EXAMPLE.COM**

Un altro esempio, questa volta usando nomi di realm i cui nomi non hanno suffissi in comune (**DEVEL.EXAMPLE.COM** e **PROD.EXAMPLE.ORG) DEVEL.EXAMPLE.COM → EXAMPLE.COM → COM → ORG → EXAMPLE.ORG → PROD.EXAMPLE.ORG**

- **DEVEL.EXAMPLE.COM** e **EXAMPLE.COM** condividono una chiave per **krbtgt/EXAMPLE.COM@DEVEL.EXAMPLE.COM**
- **EXAMPLE.COM** e **COM** condividono una chiave per **krbtgt/COM@EXAMPLE.COM**
- **COM** e **ORG** condividono una chiave per **krbtgt/ORG@COM**
- **ORG** e **EXAMPLE.ORG** condividono una chiave per **krbtgt/EXAMPLE.ORG@ORG**
- **EXAMPLE.ORG** e **PROD.EXAMPLE.ORG** condividono una chiave per **krbtgt/PROD.EXAMPLE.ORG@EXAMPLE.ORG**

Il metodo più complicato ma anche più flessibile, comporta la configurazione della sezione **capaths** nel file **/etc/krb5.conf**, permettendo ai client che hanno le credenziali per un realm di trovare il realm successivo nella catena, che eventualmente li autenticherà al server.

L'interpretazione della sezione **capaths** è relativamente immediato: la voce iniziale nella sezione è il nome del realm in cui si trova il client. All'interno della sezione, si trovano elencati i realm intermedi, da cui il client deve ottenere le credenziali. Se non ci sono realm intermedi, si usa il valore ".".

Ecco un esempio:

```
[capaths]
A.EXAMPLE.COM = {
B.EXAMPLE.COM = .
C.EXAMPLE.COM = B.EXAMPLE.COM
D.EXAMPLE.COM = B.EXAMPLE.COM
D.EXAMPLE.COM = C.EXAMPLE.COM
}
```

Nell'esempio, i client nel realm **A.EXAMPLE.COM** possono ottenere le credenziali cross-realm per **B.EXAMPLE.COM**, direttamente dal KDC del realm **A.EXAMPLE.COM**.

Se quei client vogliono contattare un servizio del realm **C.EXAMPLE.COM**, essi devono prima ottenere le credenziali dal realm **B.EXAMPLE.COM** (occorre che esista **krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM**), e poi usare **queste** credenziali, per ottenere le credenzialli da usare nel realm **C.EXAMPLE.COM** (usando **krbtgt/C.EXAMPLE.COM@B.EXAMPLE.COM**).

Se quei client vogliono contattare un servizio del realm **D.EXAMPLE.COM**, essi devono prima ottenere le credenziali dal realm **B.EXAMPLE.COM**, e poi quelle dal realm **C.EXAMPLE.COM**, prima di ottenere finalmente le credenziali da usare con il realm **D.EXAMPLE.COM**.

Nota

Senza una sezione capath che indichi il contrario, Kerberos assume che la relazione di fiducia cross-realm, sia di tipo gerarchico.

I client nel realm **A.EXAMPLE.COM** possono ottenere credenziali cross-realm, direttamente dal realm **B.EXAMPLE.COM**. Senza l'indicazione del ".", il client avrebbe provato ad usare una ricerca di tipo gerarchico; in questo caso:

```
A.EXAMPLE.COM → EXAMPLE.COM → B.EXAMPLE.COM
```

2.6.10. Ulteriori risorse

Per maggiori informazioni su Kerberos, fare riferimento alle seguenti risorse.

2.6.10.1. Documentazione locale su Kerberos

- *Kerberos V5 Installation Guide* e *Kerberos V5 System Administrator's Guide*, in formato PostScript ed HTML. Le guide si trovano nella directory **/usr/share/doc/krb5-server-<*version-number*>/**, dove <*version-number*> è la version del pacchetto **krb5-server** installato.

- *Kerberos V5 UNIX User's Guide*, in formato PostScript ed HTML. La guida si trova nella directory **/usr/share/doc/krb5-workstation-<*version-number*>/**, in cui <*version-number*> è la versione del pacchetto **krb5-workstation** installato.

- Pagine di man relative a Kerberos — Ci sono un buon numero di pagine man, che descrivono le varie applicazioni e i file di configurazione riguardanti una implementazione di Kerberos. Di seguito, si riporta un elenco delle più importanti pagine di man.

  Applicazioni Client

  - **man kerberos** — Una introduzione al sistema Kerberos, in cui viene descritto come funzionano le credenziali, oltre a utili raccomandazioni su come ottenere e distruggere i ticket emessi da Kerberos. La parte finale della pagina di man, contiene i riferimenti ad ulteriori pagine.

  - **man kinit** — Descrive come usare questo comando per ottenere e memorizzare i ticket.

  - **man kdestroy** — Descrive come usare questo comando per distruggere le credenziali Kerberos.

  - **man klist** — Descrive come usare questo comando per visualizzare le credenziali Kerberos memorizzate.

Applicazioni Amministrative

- **`man kadmin`** — Descrive come usare questo comando per amministrare il database Kerberos V5.
- **`man kdb5_util`** — Descrive come usare questo comando per creare ed effettuare operazioni amministrative di basso livello, sul database Kerberos V5.

Applicazioni Server

- **`man krb5kdc`** — Descrive le opzioni disponibili da riga di comando per il KDC Kerberos V5.
- **`man kadmind`** — Descrive le opzioni disponibili da riga di comando per l'AS Kerberos V5.

File di Configurazione

- **`man krb5.conf`** — Descrive il formato e le opzioni disponibili, nel file di configurazione, per la libreria Kerberos V5.
- **`man kdc.conf`** — Descrive il formato e le opzioni disponibili, nel file di configurazione, per l'AS e il KDC Kerberos V5.

2.6.10.2. Siti utili su Kerberos

- *Kerberos: The Network Authentication Protocol*[13] — sul sito del MIT.
- *The Kerberos Frequently Asked Questions* [14] — Utili Domande/Risposte su Kerberos
- *Kerberos: An Authentication Service for Open Network Systems*[15] — E' la versione PostScript del documento originario su Kerberos, scritto da Jennifer G. Steiner, Clifford Neuman, e Jeffrey I. Schiller.
- *Designing an Authentication System: a Dialogue in Four Scenes*[16] — Questo documento, scritto originariamente da Bill Bryant nel 1988, e modificato da Theodore Ts'o nel 1997, è una conversazione tra due sviluppatori che riflettono sul progetto di un sistema di autenticazione in stile Kerberos. Lo stile colloquiale della discussione, lo rende un buon punto di partenza per coloro che sono completamente all'oscuro di Kerberos.
- *How to Kerberize your site*[17] — E' un buon riferimento per kerberizzare una rete.
- *Kerberos Network Design Manual*[18] — Fornisce una panoramica sul sistema Kerberos.

2.7. Firewall

La sicurezza nell'informazione comunemente è visto come un processo e non come un prodotto. Le implementazioni volte a garantire una sicurezza standard, solitamente impiegano dei meccanismi per il controllo degli accessi e per limitare le risorse di rete solo agli utenti autorizzati, identificabili e tracciabili. Fedora include molti strumenti per amministratori e ingegneri addetti alla sicurezza, utili per controllare gli accessi in ambito di rete.

[13] http://web.mit.edu/kerberos/www/
[14] http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html
[15] ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS
[16] http://web.mit.edu/kerberos/www/dialogue.html
[17] http://www.ornl.gov/~jar/HowToKerb.html
[18] http://www.networkcomputing.com/netdesign/kerb1.html

I firewall, figurano tra i componenti di base per una implementazione di rete sicura. Molti produttori di firewall commerciali, forniscono soluzioni per ogni livello di necessità: dai firewall per proteggere i PC di utenti domestici a quelli dedicati ai centri di elaborazioni dati. I firewall possono essere hardware a sè stanti, come i dispositivi realizzati da Cisco, Nokia e Sonicwall, oppure soluzioni software, come i firewall sviluppati da Checkpoint, McAfee e Symantec, per il mercato casalingo e aziendale.

Oltre alla differenza fra firewall hardware e software, i firewall si distinguono anche nel loro modo di funzionare. La *Tabella 2.2, «Tipi di firewall»* illustra tre tipi comuni di firewall e il loro funzionamento:

Tabella 2.2. Tipi di firewall

Metodo	Descrizione	Vantaggi	Svantaggi
NAT	*NAT* (Network Address Translation), posiziona le sottoreti private dietro unico indirizzo IP pubblico o un limitato gruppo di indirizzi IP pubblici, mascherando tutte le richieste verso un'unica destinazione. Il kernel Linux presenta funzionalità NAT integrate tramite il sottosistema Netfilter.	· Può essere configurato in modo trasparente alle macchine sulla LAN · La protezione di macchine e servizi dietro uno o più indirizzi IP (esterni) semplifica i compiti di amministrazione · Gli accessi in ingresso e in uscita dalla LAN possono essere configurati aprendo e chiudendo le porte sul firewall/gateway NAT	· Impossibile prevenire attività maliziose da parte di connessioni esterne al firewall
Filtro dei pacchetti	Un firewall di filtraggio dei pacchetti analizza tutti i pacchetti che passano attraverso la LAN. Può leggere e analizzare i pacchetti in base alle informazioni di intestazione, e filtrare i pacchetti secondo un insieme di regole programmabili implementate dall'amministratore. Il kernel Linux presenta funzionalità di filtraggio in modo nativo attraverso il sottosistema Netfilter.	· Configurabile attraverso l'utlity **`iptables`** · Non richiede nessuna configurazione sul lato client, poichè tutta l'attività di rete viene filtrata a livello router e non a livello applicazione · Poichè i pacchetti non vengono trasmessi attraverso un proxy, le prestazioni di rete risultano più elevate grazie alla connessione diretta tra client ed host remoto	· Impossibile filtrare i pacchetti per contenuto come avviene con un firewall proxy · L'analisi dei pacchetti è a livello protocollo di trasmissione e non a livello applicazione · Architetture di rete complesse possono rendere ardua la stesura delle regole di filtraggio, specialmente se combinate con *mascheramento IP* o sottoreti locali e con reti DMZ
Proxy	I firewall proxy filtrano tutte le richieste di un certo protocollo o tipo, dai client LAN ad una macchina proxy, che a nome del client le trasmette su Internet. Una macchina proxy agisce come un buffer fra utenti remoti maliziosi e i client della rete interna.	· E' possibile controllare le applicazione e i protocolli in funzione all'esterno della LAN · Alcuni server proxy mantengono una copia locale dei dati richiesti frequentemente invece di richiederli ogni volta su Internet. Ciò aiuta a ridurre il consumo di banda · I servizi proxy possono registrare su file la	· I proxy spesso sono implementati per applicazioni specifiche (HTTP, Telnet, ecc.), oppure limitati ad un protocollo (la maggior parte dei proxy funziona solo con servizi TCP) · Le applicazioni server non funzionano con i proxy, quindi per queste occorre usare una diversa forma di sicurezza

Metodo	Descrizione	Vantaggi	Svantaggi
		loro attività (logging), permettendo un monitoraggio/controllo maggiore sull'utilizzo delle risorse di rete	· I proxy possono diventare dei colli di bottiglia, in quanto tutto il traffico deve passare attraverso un intermediario

2.7.1. Netfilter e IPTables

Il kernel Linux fornisce un potente sottosistema di rete chiamato *Netfilter*. Netfilter è in grado di fornire filtraggio stateful o stateless, servizi NAT e mascheramento degli indirizzi IP. Inoltre può *alterare* le informazioni di intestazione dei pacchetti IP per il routing avanzato e gestire lo stato della connessione. Netfilter è controllato con lo strumento **`iptables`**.

2.7.1.1. Panoramica su IPTables

La forza e la flessibilità di Netfilter si avvale di **`iptables`**, uno strumento da terminale simile nella sintassi, al suo predecessore, **`ipchains`**, sostituito da Netfilter/iptables a partire dal kernel 2.4.

`iptables` usa Netfilter per migliorare la connessione, l'ispezione e l'analisi della rete. Le caratteristiche di **`iptables`** includono in una unica interfaccia da linea di comando logging avanzato, azioni *pre- e post-routing*, *network address translation* e *port forwarding*.

Questa sezione ha dato solo una breve descrizione di **`iptables`**. Per informazioni più dettagliate, fare riferimento alla *Sezione 2.8, «IPTables»*.

2.7.2. Configurazione di un firewall di base

Così come in una costruzione medioevale il muro tagliafuoco tenta di prevenire la propagazione del fuoco, il firewall di un computer tenta di impedire che software maliziosi si propaghino nel computer. Un firewall serve anche ad impedire che utenti non autorizzati possano accedere al computer.

In una installazione predefinita di Fedora esiste un firewall tra il proprio computer (o la rete locale), e una qualsiasi rete non sicura come ad esempio Internet. Esso imposta i servizi ai quali possono accedere gli utenti remoti. Un firewall correttamente configurato, può incrementare notevolmente la sicurezza del sistema. Si raccomanda di configurare un firewall su tutti i sistemi Fedora con una connessione ad internet.

2.7.2.1. Srtumento di Amministrazione Firewall

Durante l'installazione di Fedora, nella schermata **Configurazione Firewall** si può abilitare un firewall di base come pure autorizzare su particolari schede di rete, servizi di ingresso e porte.

Dopo l'installazione, è possibile cambiare queste preferenze utilizzando lo strumento **Amministrazione Firewall**.

Per avviare l'applicazione, usare il seguente comando:

```
[root@myServer ~] # system-config-firewall
```
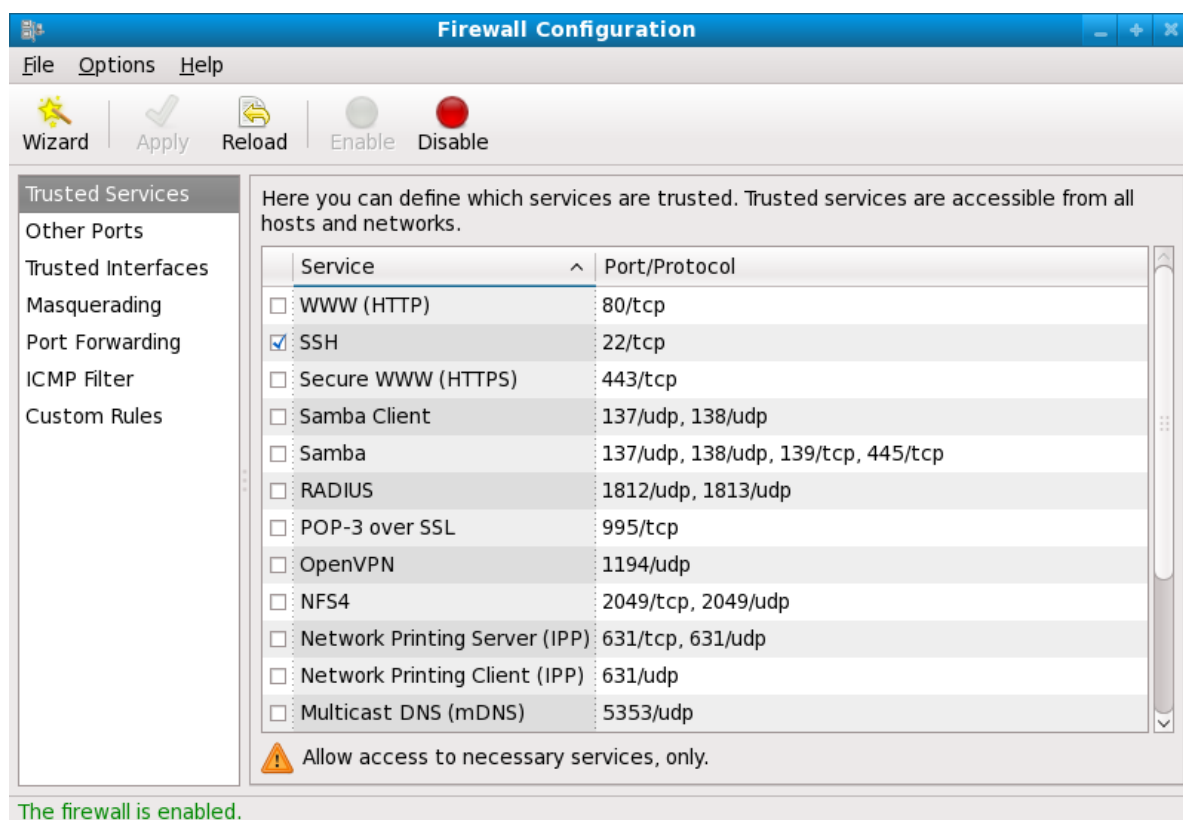
Figura 2.10. **Srtumento di Amministrazione Firewall**

Nota

Amministrazione Firewall configura solo un firewall di base. Se il sistema necessita di regole più complesse, fare riferimento alla *Sezione 2.8, «IPTables»* contenente i dettagli sulla configurazione di regole **`iptables`**.

2.7.2.2. Abilitare e disabilitare il firewall

Selezionare una delle seguenti opzioni per il firewall:

- **Disabilitato** — Questa opzione consente il completo accesso al sistema, privando il sistema di ogni controllo di sicurezza. Usare questa impostazione soltanto se il sistema si trova in una rete sicura (senza connessione ad Internet), o se si configura un firewall personalizzato, utilizzando lo strumento da linea di comando **`iptables`**.

Avviso

Le configurazioni e le regole personalizzate del firewall sono salvate nel file **`/etc/sysconfig/iptables`**. Se si seleziona **Disabilitato** e si preme **OK** le attuali configurazioni e regole di firewall vengono azzerate.

- **Abilitato** — Questa opzione configura il sistema a rifiutare le richieste di connessioni in ingresso, ossia tutte quelle connessioni provenienti dall'esterno che non corrispondono a richieste effettuate dal sistema, come repliche DNS o richieste DHCP. Se occorre autorizzare l'accesso a servizi in esecuzione sulla macchina, essi possono essere impostati nel firewall.

  Se il sistema è collegato ad Internet ma non esegue alcun server, questa opzione è la scelta più sicura.

2.7.2.3. Servizi fidati

Abilitando le opzioni nella lista **Servizi fidati**, si autorizza il servizio a passare attraverso (bypass) il firewall.

WWW (HTTP)

Il protocollo HTTP è usato da Apache (e da altri server web) per servire pagine web. Se si intende rendere pubblico il proprio server web, abilitare la check-box relativa. Non occorre abilitare questa opzione per visualizzare pagine web sul server locale o per lo sviluppo di pagine web. Questo servizio richiede che sia installato il pacchetto **`httpd`**.

L'abilitazione di **WWW (HTTP)** non apre una porta per il servizio HTTPS, la versione SSL di HTTP. Se è necessario questo servizio, abilitare la check-box relativa al server **Secure WWW (HTTPS)**.

FTP

Il protocollo FTP è usato per trasferire file fra computer. Se si intende creare un server FTP disponibile pubblicamente, abilitare la check-box relativa. Questo servizio richiede che sia installato il pacchetto **`vsftpd`**.

SSH

SSH (Secure Shell) è una raccolta di strumenti per accedere ed eseguire comandi su una macchina remota. Per autorizzare l'accesso remoto alla macchina via ssh, abilitare la check-box relativa. Questo servizio richiede che sia installato il pacchetto **`openssh-server`**.

Telnet

Telnet è un protocollo per accedere a macchine remote. Le comunicazioni Telnet non sono cifrate e non offrono nessuna protezione contro le intercettazioni. Consentire l'accesso Telnet in ingresso non è raccomandato. Per autorizzare l'accesso alla macchina via Telnet, abilitare la check-box relativa. Questo servizio richiede che sia installato il pacchetto **`telnet-server`**.

Mail (SMTP)

SMTP è un protocollo che consente ad host remoti di connettersi direttammente ad una macchina per l'invio di mail. Non si deve abilitare questo servizio se si riceve la posta dal proprio ISP, via POP3 o IMAP oppure se si utilizza uno strumento come **`fetchmail`**. Per consentire la consegna di posta dalla macchina remota abilitare questa check-box. Notare che un server SMTP configurato in modo scorretto, potrebbe consentire a macchine remote di usare il server per l'invio di spam.

NFS4

NFS (Network File System) è un protocollo di condivisione file usato comunemente sui sistemi *NIX. La versione 4 di questo protocollo è più sicuro dei suoi predecessori. Se si desidera condividere i propri file o cartelle con altri utenti della rete, abilitare questa check-box.

Samba

Samba è una implementazione del protocollo di rete proprietario, SMB. Se si desidera condividere file, cartelle o stampanti locali con macchine microsoft windows, abilitare questa check-box.

2.7.2.4. Altre porte

Lo strumento di **Amministrazione Firewall** include una sezione **Altre porte** per impostare in **iptables** i numeri delle porte IP fidate. Per esempio, per permettere ad IRC ed IPP (Internet Printing Protocol) di superare le regole del firewall, aggiungere quanto segue alla sezione **Altre porte**:

194:tcp,631:tcp

2.7.2.5. Salvare le impostazioni

Premere il pulsante **OK** per salvare i cambiamenti apportati al firewall. Se è stato selezionato **Abilita firewall**, le opzioni selezionate verranno tradotte in comandi **iptables** e salvate nel file **/etc/sysconfig/iptables**. Immediatamente dopo il salvataggio, viene ri-avviato automaticamente il servizio **iptables** in modo da rendere immediate le modifiche apportate al firewall. Se invece è stato selezionato **Disabilita firewall**, il file **/etc/sysconfig/iptables** viene eliminato ed il servizio **iptables** immediatamente interrotto.

Comunque, le varie impostazioni vengono salvate anche nel file **/etc/sysconfig/system-config-firewall**, usato dal sistema al successivo riavvio dell'applicazione per il regolare ripristino delle impostazioni. Si raccomanda di non modificare direttamente questo file.

Anche se il filrewall viene avviato immediatamente, il servizio **iptables** non è configurato per avviarsi automaticamente al boot. Per maggiori informazioni, fare riferimento alla *Sezione 2.7.2.6, «Attivare il servizio IPTables»*.

2.7.2.6. Attivare il servizio IPTables

Le regole del firewall sono attive solo se **iptables** è in esecuzione. Per avviare manualmente il servizio, usare il seguente comando:

```
[root@myServer ~] # service iptables restart
```

Per far sì che **iptables** si avvii al boot, usare il seguente comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

2.7.3. Usare IPTables

Il primo passo da fare per utilizzare **iptables**, è avviare il servizio **iptables**. Usare il seguente comando per avviare il servizio **iptables**:

```
[root@myServer ~] # service iptables start
```

Nota

Il servizio **ip6tables** può essere disabilitato se si usa solo il servizio **iptables**. Se si disattiva il servizio **ip6tables**, ricordarsi di disattivare anche la rete IPv6. Non lasciare mai attivo un dispositivo di rete, senza il firewall corrispondente.

Per avviare **iptables** al boot di sistema, usare il seguente comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

In tal caso, **`iptables`** si avvia automaticamente nei runlevel 3, 4 o 5.

2.7.3.1. Sintassi del comando iptables

Il seguente esempio, illustra la sintassi di base del comando **`iptables`**:

```
[root@myServer ~ ] # iptables -A <chain> -j <target>
```

L'opzione **`-A`** specifica che la regola deve essere aggiunta alla <*chain*> (catena). Ogni catena è costituita da una o più *rules* (regole) ed è perciò meglio nota come una *ruleset* (insieme di regole).

Le tre catene preesistenti sono INPUT, OUTPUT e FORWARD. Queste catene sono permanenti e non possono essere eliminate. La catena specifica il punto in cui il pacchetto viene manipolato.

L'opzione **`-j <target>`** (obbiettivo), specifica un'azione ossia cosa fare se il pacchetto corrisponde alla regola. Esempi di target predefiniti sono ACCEPT, DROP e REJECT.

Per maggiori informazioni su catene, opzioni e target disponibili, fare riferimento alle pagine di man su **`iptables`**.

2.7.3.2. Policy di base

Stabilire una policy per il firewall di base serve da fondamenta su cui costruire delle regole più dettagliate.

Ogni catena di **`iptables`** è costituita da una policy predefinita e da zero o più regole che complessivamente definiscono le regole per il firewall.

La policy predefinita di una catena può essere DROP o ACCEPT. Gli amministratori accorti di solito implementano una policy predefinita di DROP e autorizzano solo particolari pacchetti, sulla base di un'analisi caso-per-caso. Per esempio, le seguenti policy bloccano tutti i pacchetti in ingresso e in uscita da un gateway:

```
[root@myServer ~ ] # iptables -P INPUT DROP
[root@myServer ~ ] # iptables -P OUTPUT DROP
```

Si raccomanda inoltre di vietare qualsiasi *forward* di pacchetti (cioè traffico di rete che deve essere re-indirizzato dal firewall al nodo di destinazione), per limitare l'esposizione involontaria ad Internet dei client interni. Per fare ciò, usare la seguente regola:

```
[root@myServer ~ ] # iptables -P FORWARD DROP
```

Una volta impostate le policy predefinite per una catena, si possono creare e salvare ulteriori regole, secondo i propri requisiti di rete e di sicurezza.

Le seguenti sezioni descrivono come salvare le regole iptables e illustrano come implementare le regole per la costruzione del proprio firewall.

2.7.3.3. Salvare e ripristinare le regole IPTables

I cambiamenti a **`iptables`** se non vengono opportunamente salvati, restano transitori: se si riavvia il sistema o se il servizio **`iptables`** viene riavviato, le regole appena create/modificate vengono

automaticamente scaricate e resettate. Per salvare le regole in modo permanente, occorre usare il seguente comando:

```
[root@myServer ~ ] # service iptables save
```

Le regole sono salvate nel file **`/etc/sysconfig/iptables`** e vengono applicate all'avvio del servizio o al riavvio della macchina.

2.7.4. Filtraggi IPTables comuni

Uno degli aspetti più importanti della sicurezza di rete è impedire l'accesso alla LAN da parte di attaccanti. L'integrità della LAN può essere garantita impostando stringenti regole di firewall.

Tuttavia, una policy impostata per bloccare tutti i pacchetti in ingresso, uscita e re-instradati, renderebbe del tutto impossibile a firewall/gateway e agli utenti interni alla LAN la comunicazione fra loro e con le risorse esterne.

Quindi gli amministratori, per consentire ai propri utenti di usufruire delle funzioni e delle applicazioni di rete, devono necessariamente aprire determinate porte alla comunicazione.

Per esempio, per consentire l'accesso alla porta numero 80 *sul firewall*, aggiungere la seguente regola:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Ciò permette agli utenti di esplorare i siti Internet che comunicano sulla porta standard numero 80. Per consentire l'accesso a siti web sicuri (per esempio, https://www.example.com/), occorre abilitare l'accesso anche attraverso la porta numero 443, come di seguito riportato:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Importante

Quando si crea un insieme di regole di **`iptables`**, l'ordine è importante.

Se una regola specifica di scartare qualsiasi pacchetto proveniente dalla sottorete 192.168.100.1/24, e questa è seguita da una regola che specifica di accettare i pacchetti provenienti dall'indirizzo 192.168.100.13 (che si trova all'interno della sottorete), allora la seconda regola viene ignorata.

Per accettare i pacchetti provenienti da 192.168.100.13, la regola relativa deve precedere la regola che scarta i pacchetti prevenienti dalla sottorete.

Per inserire una regola in una specifica posizione di una catena esistente, usare l'opzione **`-I`**. Per esempio:

```
[root@myServer ~ ] # iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

Questa è la prima regola nella catena INPUT ed autorizza il traffico di loopback sul dispositivo.

Per accedere ai servizi remoti di una LAN si possono usare servizi sicuri come SSH che impiegano connessioni cifrate.

Nel caso di risorse basate su PPP (come modem o router ISP), si usano accessi dial-up per circuire le barriere del firewall. Trattandosi di connessioni dirette, le connessioni via modem tipicamente si trovano dietro un firewall/gateway.

Per utenti con connessioni a banda larga, comunque, si presentano dei casi particolari. Si può configurare **iptables** in modo da accettare connessioni via SSH. Per esempio, le seguenti regole consentono l'accesso remoto via SSH:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@myServer ~ ] # iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Queste due regole autorizzano l'accesso in entrata e in uscita da un nodo, quale può essere un PC connesso direttamente ad Internet o un firewall/gateway, ma impediscono l'accesso al servizio ai nodi dietro al firewall/gateway. Per consentire a tutta la LAN di accedere a questo servizio, si potrebbe usare un NAT (*Network Address Translation*) insieme a regole di filtraggio, **iptables**.

2.7.5. Regole di FORWARD e NAT

La maggior parte dei provider ISP offrono, ai propri clienti, solo un numero limitato di indirizzi pubblici IP.

Per questo motivo, gli amministratori devono disporre di un metodo che senza far uso di indirizzi IP pubblici, consenta ai nodi della LAN di accedere ai servizi Internet ed il metodo più comune consiste nell'usare indirizzi IP privati.

I router di soglia (come i firewall) ricevono da Internet le trasmissioni in ingresso e re-indirizzano i pacchetti al nodo LAN interessato. Allo stesso modo, i firewall/gateway possono anche re-indirizzare le richieste in uscita, da un nodo LAN al servizio Internet remoto.

Questo re-indirizzamento del traffico di rete, a volte, potrebbe diventare una minaccia, specialmente con l'alta disponibilità dei moderni strumenti di cracking, in grado di *imitare* gli indirizzi IP *interni*, mascherando la macchina remota dell'attaccante come un nodo della LAN.

Per impedire tutto ciò, **iptables** fornisce policy di routing e di forwarding (instradamento e re-indirizzamento), che se adeguatamente implementate impediscono un uso anormale delle risorse di rete.

La catena **FORWARD** consente ad un amministratore di controllare il routing dei pacchetti all'interno della LAN. Per esempio, per consentire il re-indirizzamento sull'intera LAN (assumendo che al firewall/gateway sia assegnato un indirizzo IP interno, associato alla scheda eth1), si possono usare le seguenti regole:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth1 -j ACCEPT
[root@myServer ~ ] # iptables -A FORWARD -o eth1 -j ACCEPT
```

Queste regole stabiliscono che i sistemi dietro al firewall/gateway possono accedere alla intera rete interna. Ossia il gateway trasferisce i pacchetti da un nodo della LAN al nodo di destinazione, passando tutti i pacchetti attraverso la scheda **eth1**.

Nota

Per impostazione, la policy IPv4 nei kernel Fedora disabilita il supporto al forwarding IP e ciò impedisce a sistemi Fedora di funzionare come router di soglia dedicati. Per abilitare il forwarding IP, usare il seguente comando:

```
[root@myServer ~ ] # sysctl -w net.ipv4.ip_forward=1
```

Questa modifica di configurazione, dura solo per la sessione corrente: non persiste dopo un riavvio della macchina o un riavvio dei servizi di rete. Per impostare permanentemente il forwarding IP, modificare il file **`/etc/sysctl.conf`** come indicato di seguito:

Individuare la seguente riga:

```
net.ipv4.ip_forward = 0
```

Modificarla come segue:

```
net.ipv4.ip_forward = 1
```

Usare il seguente comando per abilitare le modifiche al file **`sysctl.conf`**:

```
[root@myServer ~ ] # sysctl -p /etc/sysctl.conf
```

2.7.5.1. Postrouting e mascheramento IP

Per ora, l'impostazione del forwarding dei pacchetti via la scheda interna del firewall, consente ai nodi delle LAN di comunicare tra di loro ma essi non possono ancora comunicare esternamente, verso Internet.

Per consentire ai nodi, con indirizzi IP privati, di comunicare con reti pubbliche esterne occorre configurare il firewall per il *mascheramento IP*, ossia mascherare le richieste provenienti dai nodi della LAN, con l'indirizzo IP della scheda di rete esterna del firewall (in questo caso, eth0):

```
[root@myServer ~ ] # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Questa regola usa la tabella di corrispondenza dei pacchetti, NAT (**`-t nat`**) e specifica sulla scheda di rete esterna (**`-o eth0`**), la catena POSTROUTING (**`-A POSTROUTING`**).

Quindi la regola POSTROUTING permette l'alterazione dell'indirizzo IP dei pacchetti mentre questi lasciano la scheda di rete esterna del firewall.

Il target **`-j MASQUERADE`** specifica di mascherare gli indirizzi IP privati con l'indirizzo IP esterno del firewall/gateway.

2.7.5.2. Prerouting

Per rendere pubblico un server delle rete interna, si può usare l'opzione **`-j DNAT`** della catena PREROUTING specificando un indirizzo IP di destinazione e un numero di porta a cui indirizzare i pacchetti in ingresso richiedenti il servizio.

Per esempio, per re-indirizzare le richieste HTTP al proprio server HTTP Apache, localizzato all'indirizzo 172.31.0.23, usare il seguente comando:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to
 172.31.0.23:80
```

Questa regola specifica che la tabella NAT usa la catena PREROUTING, re-indirizzando le richieste HTTP in ingresso, esclusivamente all'indirizzo IP 172.31.0.23.

Nota

Se nella catena FORWARD è presente una policy predefinita di DROP, perchè il mascheramento IP sia possibile, occorre inserire in coda una regola di forward che re-indirizzi tutte le richieste HTTP. Per fare ciò, usare il seguente comando:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 172.31.0.23 -j
 ACCEPT
```

Questa regola re-indirizza tutte le richieste HTTP dal firewall al server HTTP Apache, dietro il firewall.

2.7.5.3. DMZ e IPTables

Si possono creare regole **`iptables`** che re-indirizzino il traffico verso macchine dedicate, come server HTTP o FTP in una rete DMZ (*demilitarized zone*). Una DMZ è una speciale sottorete locale, dedicata quasi esclusivamente a fornire servizi verso reti pubbliche come Internet.

Per esempio, per impostare una regola di re-indirizzamento, che instradi le richieste HTTP in ingresso verso un server HTTP dedicato su 10.0.4.2 (fuori dal range della LAN 192.168.1.0/24), si potrebbe usare la seguente regola di **`PREROUTING`**:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-
destination 10.0.4.2:80
```

Con questo comando, tutte le connessioni HTTP diretta alla porta 80 vengono instradate verso il server HTTP della sottorete DMZ. Questo tipo di segmentazione della rete si dimostra molto più sicuro, rispetto a connessioni HTTP dirette ad una macchina nella rete LAN interna.

Se il server HTTP è configurato per accettare connessioni sicure, allora si dovrà re-instradare anche la porta 443.

2.7.6. Software maliziosi e indirizzi IP spoofed

Con **`iptables`** si possono creare regole anche più elaborate per controllare l'accesso a specifiche sottoreti o anche a particolari nodi della LAN. E si può anche impedire che applicazioni o programmi sospetti, come trojan, worm e altri virus client/server contattino i loro server.

Per esempio, alcuni trojan scansionano la rete alla ricerca di servizi attivi nel range di porte tra 31337 e 31340 (chiamate porte *elite* nel gergo cracker).

Dato che non esistono servizi legittimati che comunicano su queste porte non standard, bloccarle serve a ridurre la possibilità che nodi potenzialmente infetti sulla LAN, possano comunicare autonomamente, con i loro server remoti.

Le seguenti regole, scartano tutto il traffico TCP che tenti di usare la porta 31337:

```
[root@myServer ~ ] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
[root@myServer ~ ] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
```

Si possono bloccare anche le connessioni esterne, che maliziosamente tentano di "imitare" (spoof) il range di indirizzi IP privati per intrufolarsi nella LAN.

Per esempio, se la LAN usa il range 192.168.1.0/24, è possibile impostare una regola sulla scheda di rete esterna (connessa ad Internet, per esempio eth0), che scarti tutti i pacchetti con indirizzi IP nel range della LAN.

Poichè per policy predefinita, si raccomanda di scartare i pacchetti re-indirizzati, qualsiasi indirizzo IP *spoofed* proveniente dal dispositivo di rete esterno (eth0), viene a maggior ragione respinto.

```
[root@myServer ~ ] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

Nota

Esiste una differenza tra **`DROP`** e **`REJECT`** quando si tratta di regole *aggiunte* in coda.

`REJECT` rifiuta l'accesso e ritorna un messaggio di **`connessione rifiutata`** agli utenti che tentano di connettersi al servizio. Il comando **`DROP`**, come lascia intendere il nome, scarta i pacchetti senza nessun messaggio.

Gli amministratori possono scegliere a propria discrezione quando usare le due opzioni. Comunque, per evitare confusione e ripetuti tentavi di connessione da parte di utenti, si raccomanda di usare l'opzione **`REJECT`**.

2.7.7. IPTables e Connection Tracking

E' possibile ispezionare e restringere l'accesso ai servizi, anche in base al loro *stato di connessione*. Un modulo all'interno di **`iptables`** usa un metodo denominato *connection tracking* (tracciamento delle connessioni), per immagazzinare informazioni sulle connessioni in ingresso. Si può consentire o rifiutare l'accesso in base ai seguenti stati di connessione:

- **`NEW`** — Un pacchetto che richiede una nuova connessione, come una richiesta HTTP
- **`ESTABLISHED`** — Un pacchetto che fa parte di una connessione esistente.
- **`RELATED`** — Un pacchetto che richiede una nuova connessione, ma che appartiene ad una connessione esistente. Per esempio, FTP usa la porta numero 21 per stabilire una connessione, ma i dati vengono trasmessi su una porta differente (tipicamente la porta 20).
- **`INVALID`** — Un pacchetto che non fa parte di nessuna connessione della connection tracking.

Le funzioni di stato di *connection tracking*, possono essere usate con qualsiasi protocollo di rete, anche con protocolli privi di stato (come UDP). Il seguente esempio mostra una regola che usa *connection tracking*, trasferendo solo i pacchetti appartenenti ad una connessione *ESTABLISHED e RELATED*:

```
[root@myServer ~ ] # iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

2.7.8. IPv6

L'introduzione del nuovo Internet Protocol di futura generazione, l'IPv6, espande la limitazione degli indirizzi a 32bit di IPv4 (o IP). IPv6, infatti, supporta indirizzi a 128bit, e le reti compatibili con IPv6, presentano perciò una maggiore capacità di indirizzamento.

Fedora supporta regole di firewall IPv6 usando Netfilter 6 e il comando **ip6tables**. In Fedora 14, sia IPv4 sia IPv6, sono abilitati in modo predefinito

La sintassi del comando **ip6tables** è identica a **iptables**, a parte il fatto che supporta indirizzi a 128bit. Per esempio, usare il seguente comando per abilitare connessioni SSH su un server di rete IPv6:

```
[root@myServer ~ ] # ip6tables -A INPUT -i eth0 -p tcp -s 3ffe:ffff:100::1/128 --dport 22 -j
 ACCEPT
```

Per maggiori informazioni sulle reti IPv6, fare riferimento alla pagina web *Welcome to the IPv6 Information Page!* [19].

2.7.9. Ulteriori risorse

Molti aspetti su firewall e Netfilter non sono stati adeguatamente esposti ed approfonditi in questo capitolo, che vuole essere una introduzione ed uno stimolo per ulteriori letture. Per chi volesse approfondire l'argomento, di seguito si riportano alcune interessanti risorse.

2.7.9.1. Documentazione installata riguardante i firewall

- Per informazioni sul comando **iptables** e le opzioni disponibili, vedere la *Sezione 2.8, «IPTables»*.
- La pagina di man su **iptables** contiene una spiegazione delle varie opzioni.

2.7.9.2. Siti utili sui firewall

- *Netfilter*[20] — Il sito ufficiale dei progetti Netfilter e **iptables**.
- *tldp.org*[21] — The Linux Documentation Project, contiene molte guide utili, relative alla creazione e all'amministrazione di un firewall.
- *Internet Assigned Numbers Authority*[22] — La lista ufficiale dei numeri di porta assegnati ai servizi, così come stabilito dall'IANA (Internet Assigned Numbers Authority).

2.7.9.3. Documentazione relativa

- *Red Hat Linux Firewalls* di Bill McCarty (Red Hat Press) — Un manuale su come costruire firewall server e di rete, usando tecnologie open source, come Netfilter e **iptables**, per operazioni di filtraggio dei pacchetti. Include anche argomenti correlati, come l'analisi dei messaggi di firewall, sviluppo di regole di firewall e la progettazione di un firewall personale, usando vari strumenti grafici.
- *Linux Firewalls* di Robert Ziegler (New Riders Press) — Un manuale con informazioni su come creare firewall, usando sia **ipchains** del kernel, sia Netfilter e **iptables**. Vengono trattati anche

[19] http://www.ipv6.org/
[20] http://www.netfilter.org/
[21] http://www.tldp.org/
[22] http://www.iana.org/assignments/port-numbers

diversi argomenti sulla sicurezza, come le questioni riguardanti l'accesso remoto e i sistemi anti-intrusione.

2.8. IPTables

In Fedora sono inclusi avanzati strumenti di *packet filtering* (filtraggio dei pacchetti) — il processo che controlla il flusso dei pacchetti nello stack di rete del kernel a partire dal loro ingresso e fino al trasferimeto al nodo di destinazione. Le versioni del kernel precedenti alla 2.4, usavano regole **`ipchains`** per filltrare i pacchetti suddividendo il filtraggio in passaggi successivi. Il kernel 2.4 ha introdotto **`iptables`** (chiamato anche *netfilter*) che è simile a **`ipchains`** ma che espande notevolmente l'analisi e il controllo sul filtraggio.

Questo capitolo deliena le basi del filtraggio dei pacchetti spiegando le varie opzioni disponibili in **`iptables`** e come preservare le regole impostate.

Per istruzioni su come creare regole con **`iptables`** e su come impostare un firewall basato su tali regole, fare riferimento alla *Sezione 2.8.6, «Ulteriori risorse»*.

Importante

Il firewall predefinito nel kernel 2.4 e successivi, si basa su **`iptables`** che non può essere usato in concomitanza con **`ipchains`**. Quindi se **`ipchains`** è attivo all'avvio del sistema, il kernel restituirà un errore indicando l'impossibilità di avviare **`iptables`**.

Le funzionalità di **`ipchains`** non vengono influenzate da questo errore.

2.8.1. Filtraggio pacchetti

Il kernel Linux usa **Netfilter** per filtrare i pacchetti, autorizzando o meno il passaggio dei pacchetti nel sistema. Questa capacità è integrata nel kernel Linux e si basa su tre *tabelle* o *liste di regole*; esse sono:

- **`filter`** — La tabella predefinita per gestire i pacchetti.
- **`nat`** — La tabella usata per alterare i pacchetti che creano una nuova connessione e usata da *NAT* (*Network Address Translation*).
- **`mangle`** — La tabella usata per tipi specifici di alterazioni sui pacchetti.

Ogni tabella ha un gruppo di *catene* predefinite che corrispondono alle azioni eseguite da **`netfilter`** sul pacchetto.

Le catene predefinite della tabella **`filter`** sono:

- *INPUT* — Si applica ai pacchetti diretti all'host.
- *OUTPUT* — Si applica ai pacchetti generati localmente.
- *FORWARD* — Si applica ai pacchetti instradati attraverso l'host.

Le catene predefinite della tabella **`nat`** sono:

- *PREROUTING* — Altera i pacchetti in arrivo.
- *OUTPUT* — Altera i pacchetti generati localmente prima di inviarli all'esterno.

- *POSTROUTING* — Altera i pacchetti prima di inviarli all'esterno.

Le catene predefinite della tabella **`mangle`** sono:

- *INPUT* — Altera i pacchetti diretti all'host.
- *OUTPUT* — Altera i pacchetti generati localmente prima di inviarli all'esterno.
- *FORWARD* — Altera i pacchetti instradati attraverso l'host.
- *PREROUTING* — Altera i pacchetti in arrivo prima di instradarli.
- *POSTROUTING* — Altera i pacchetti prima di inviarli all'esterno.

Ogni pacchetto ricevuto o inviato da un sistema Linux è controllato da almeno una tabella ed un pacchetto, prima di emergere dalla fine della catena, viene controllato dalle regole presenti nella tabella. Ogni regola ha il proprio formato e scopo, ma generalmente tutte con l'obiettivo di identificare il pacchetto e il particolare protocollo o servizio di rete e la sua provenienza o destinazione.

Nota

Per impostazione, le regole di firewall sono salvate nei file **`/etc/sysconfig/iptables`** o **`/etc/sysconfig/ip6tables`**.

Al boot di un sistema Linux, il servizio **`iptables`** viene avviato prima di ogni servizio di DNS. Ciò significa che le regole di firewall possono riferirsi solo a indirizzi IP numerici (per esempio 192.168.0.1). Quindi eventuali nomi di dominio come host.example.com sono destinati inevitabilmente a sollevare errori.

Quando un pacchetto viene intercettato o corrisponde ad una regola di una tabella, il sistema di packet filtering applica al pacchetto un *target* o azione. Se la regola specifica un target (azione) **`ACCEPT`**, il pacchetto salta il resto dei controlli ed è autorizzato a proseguire verso la sua destinazione. Se la regola specifica un target (azione) **`DROP`**, il pacchetto viene scartato senza inviare alcuna risposta all'host mittente. Se la regola specifica un'azione **`QUEUE`**, il pacchetto è trasferito nello spazio utente. Se una regola specifica l'azione (opzionale) **`REJECT`**, il pacchetto viene scartato e all'host mittente viene risposto con un messaggio di errore.

Ogni catena ha una policy predefinita per i target (azioni) **`ACCEPT`**, **`DROP`**, **`REJECT`** e **`QUEUE`**. Se in una catena non esiste nessuna regola che si può applicare ad un pacchetto allora il pacchetto è soggetto alla policy predefinita.

Il comando **`iptables`** serve a configurare queste tabelle e all'occorrenza ad impostarne di nuove.

2.8.2. Opzioni di comando di IPTables

Le regole di filtraggio dei pacchetti si creano con il comando **`iptables`**. I seguenti aspetti di ogni pacchetto sono spesso usati come criterio:

- *Packet Type* — Specifica il tipo di pacchetti da filtrare.
- *Packet Source/Destination* — Specifica i pacchetti da filtrare in base alla loro sorgente o destinazione.
- *Target* — Specifica il target (azione) da prendere sui pacchetti corrispondenti al criterio precedente.

Per maggiori informazioni su questi criteri, vedere la *Sezione 2.8.2.4, «Match Option»* e la *Sezione 2.8.2.5, «Opzioni target»*.

Le opzioni usate con le regole di **iptables** devono essere raggruppate in modo logico, in base allo scopo e alle condizioni della regola complessiva. Il resto di questa sezione spiega le opzioni più comuni usate con il comando **iptables**.

2.8.2.1. Struttura dei comandi iptabes

Molti comandi **iptables** hanno la seguente struttura:

```
iptables [-t <table-name>] <command> <chain-name> \ <parameter-1> <option-1> \ <parameter-n> <option-n>
```

<*table-name*> — Specifica la tabella a cui applicare la regola. Se non specificata si usa la tabella **filter**.

<*command*> — Specifica l'azione da eseguire, come concatenare o eliminare una regola.

<*chain-name*> — Specifica la catena da modificare, creare o eliminare.

<*parameter*>-<*option*> — Parametri e relative opzioni che specificano come processare un pacchetto.

La lunghezza e la complessità di un comando **iptables** possono variare notevolmente, a seconda della situazione.

Per esempio, un comando per rimuovere una regola da una catena può essere molto corto:

iptables -D <*chain-name*> <*line-number*>

Al contrario, un comando che aggiunge una regola con una varietà di parametri e opzioni per filtrare i pacchetti di una sottorete, può risultare piuttosto lungo. Quando si costruiscono comandi **iptables** è importante ricordare che alcuni parametri e opzioni possono richiedere ulteriori parametri e opzioni. Ciò produce un tipico effetto cascata: parametri che richiedono ulteriori parametri. Quindi perchè la regola costruita sia valida, occorre che ogni parametro e opzione della catena sia interamente soddisfatto.

Digitare **iptables -h** per visualizzare un elenco completo delle strutture dei comandi **iptables**.

2.8.2.2. Opzioni di Comando

Le opzioni di comando indicano ad **iptables** di eseguire un'azione. In un comando **iptables** è permesso specificare solo una opzione di comando e, ad eccezione di help, deve essere espressa in caratteri maiuscoli.

Le opzioni di comando di **iptables**, sono:

- **-A** — Appende la regola alla fine della catena. Diversamente dall'opzione **-I** (descritta più avanti), non accetta alcun numero intero ma appende la regola sempre alla fine della catena.
- **-C** — Controlla una regola prima di aggiungerla alla catena. Questo comando serve a costruire regole **iptables** complesse, richiedendo interattivamente l'inserimento di parametri e opzioni.
- **-D <integer> | <rule>** — Elimina una regola da una catena usando un numero (p.e. **5** sta per la quinta regola nella catena) o specificando la regola. Quest'ultima deve corrispondere esattamente con una regola esistente.

- **`-E`** — Rinomina una catena definita dall'utente, ossia una catena non predefinita. (Fare riferimento all'opzione **`-N`** per maggiori informazioni sulle catene definite dall'utente). Si tratta di una variazione estetica senza effetti sulla struttura della tabella.

Nota

Se si tenta di rinominare una catena predefinita, il sistema restituisce l'errore **`Match not found`** (Corrispondenza non trovata): non si possono rinominare le catene predefinite.

- **`-F`** — Scarica la catena selezionata eliminando di conseguenza tutte le regole nella catena. Se non si specifica nessuna catena, questo comando scarica tutte le regole da tutte le catene.

- **`-h`** — Fornisce un elenco delle strutture dei comandi di **`iptables`** insieme ad un breve sommario dei parametri e delle opzioni disponibili.

- **`-I [<integer>]`** — Inserisce la regola nel punto specifico della catena definito dal numero. Se non viene specificato nessun numero, la regola viene inserita in cima alla catena.

Importante

Come già notato, l'ordinamento delle regole in una catena determina le regole da applicare ai pacchetti e cio è da tener presente quando si aggiunge una regola con l'opzione **`-A`** o con l'opzione **`-I`**.

Con l'opzione **`-I`** specificando il numero di un posto esistente, **`iptables`** inserisce la nuova regola *prima* della regola esistente.

- **`-L`** — Elenca tutte le regole della catena. Per elencare le regole in tutte le catene della tabella predefinita **`filter`**, non specificare alcuna catena o tabella. Invece, per elencare le regole in una catena specifica di una particolare tabella, usare la seguente sintassi:

```
iptables -L <chain-name> -t <table-name>
```

  Per maggiori informazioni sull'opzione di comando **`-L`** (in grado di visualizzare numeri di regola e descrizioni più dettagliate sulle regole), fare riferimento alla *Sezione 2.8.2.6, «Elencare le opzioni»*.

- **`-N`** — Crea una nuova catena. Il nome della catena deve essere unico altrimenti si ha un messaggio di errore.

- **`-P`** — Imposta la policy predefinita sulla catena, ossia applica il *target* (azione) specificato, per esempio ACCEPT o DROP ai pacchetti per i quali non esiste una regola corrispondente.

- **`-R`** — Sostituisce una regola nella catena. Il numero di regola deve essere specificato dopo il nome della catena. La prima regola in una catena corrisponde alla regola numero uno.

- **`-X`** — Elimina una catena precedentemente creata. Non è possibile eliminare le catene predefinite.

- **-Z**·— Imposta a zero, in tutte le catene di una tabella, i contatori di byte e di pacchetti.

2.8.2.3. Opzioni di Parametro

Per costruire una regola, alcuni comandi **iptables** inclusi quelli usati per aggiungere, appendere, eliminare, inserire o sostituire le regole in una catena, richiedono vari parametri.

- **-c** — Resetta i contatori di una regola. Questo parametro accetta le opzioni **PKTS** e **BYTES** per specificare il contatore da resettare.
- **-d** — Imposta l'hostname, l'indirizzo IP o la rete di destinazione di un pacchetto intercettato dalla regola. Nel caso di reti sono supportati i seguenti formati di indirizzo (IP/netmask) :
  - ***N.N.N.N/M.M.M.M*** — Dove *N.N.N.N* è il range di indirizzi IP e *M.M.M.M* è la netmask.
  - ***N.N.N.N/M*** — Dove *N.N.N.N* è il range di indirizzi IP e *M* è la bitmask.
- **-f** — Applica la regola solo ai pacchetti frammentati.

  Per applicare la regola solo ai pacchetti non frammentati (n.d.t. i complementari), si può usare il carattere punto esclamativo (**!**) dopo il parametro.

Nota

La tecnica della frammentazione dei pacchetti è uno standard minore del protocollo IP.

Originariamente progettato per consentire ai pacchetti IP di attraversare le reti in frame di diverse lunghezze, oggigiorno la frammentazione è usata molto spesso per generare attacchi DoS. Inoltre è importante notare che IPv6 non consente affatto la frammentazione.

- **-i** — Imposta la scheda di rete di ingresso (p.e. **eth0** o **ppp0**). Con la tabella **filter** questo parametro può essere usato solo con le catene INPUT e FORWARD; con le tabelle **nat** e **mangle** solo con la catena PREROUTING.

  Supporta anche le seguenti opzioni:

  - Punto esclamativo (**!**) — Inverte la direttiva escludendo dalla regola le interfacce specificate.
  - Somma (**+**) — Un carattere "jolly" usato per individuare tutte le interfacce che coincidono con la stringa specificata. Per esempio, il parametro **-i eth+** applicherà la regola a tutte le schede Ethernet escludendo le altre, come **ppp0**.

  Se l'opzione **-i** non ha argomento allora la regola si applica a tutte le interfacce presenti.

- **-j** — Salta al target (azione) specificato se il pacchetto è intercettato dalla regola.

  I target standard sono **ACCEPT**, **DROP**, **QUEUE**, e **RETURN**.

  Nei moduli di **iptables** caricati per default, sono disponibili anche opzioni Target Extension. Tra questi sono inclusi **LOG**, **MARK** e **REJECT**, tra gli altri. Per maggiori informazioni su questi e altri target fare riferimento alle pagine di man di **iptables**.

Questa opzione può essere usata anche per dirigere un pacchetto intercettato verso un'altra catena esterna differente, contenente altre regole da applicare al pacchetto.

Se non è specificato alcun target, il pacchetto avanza senza subire alcuna azione ed il contatore di questa regola viene incrementato di uno.

- **`-o`** — Imposta la scheda di rete di uscita. Questa opzione si applica solo alle catene OUTPUT e FORWARD della tabella **`filter`** e alla catena POSTROUTING delle tabelle **`nat`** e **`mangle`**. L'opzione accetta gli stessi parametri dell'opzione **`-i`** (che specifica la scheda di ingresso).
- **`-p <protocol>`** — Imposta il protocollo IP. Alcuni valori possibili sono **`icmp`**, **`tcp`**, **`udp`** o **`all`** oppure un valore numerico corrispondente. Più in generale si può usare un qualsiasi protocollo elencato nel file **`/etc/protocols`**.

  Il valore "**`all`**" applica la regola a tutti i protocolli supportati ed è il valore predefinito, se una regola non specifica alcun protocollo.
- **`-s`** — Imposta il mittente su un pacchetto usando la stessa sintassi dell'opzione destinazione (**`-d`**).

2.8.2.4. Match Option

Per vari protocolli di rete esistono delle match option (o opzioni di corrispondenza), configurabili per creare regole per protocolli specifici. Per usare queste opzioni occorre specificare il tipo di protocollo nel comando **`iptables`**. Per esempio, **`-p <protocol-name>`** applica le opzioni al protocollo specificato. Notare che è possibile usare anche l'ID di protocollo. Per esempio, le due regole seguenti hanno lo stesso significato:

```
iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
```

```
iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

Le definizioni dei vari servizi si trovano nel file **`/etc/services`**. Per ragioni di leggibilità, si raccomanda di usare il nome invece del numero di porta del servizio corripondente.

Avviso

Proteggere il file **`/etc/services`** da modifche non autorizzate. Se il file è modificabile, i cracker possono usare il file per abilitare le porte. Per proteggere il file, digitare come root i seguenti comandi:

```
[root@myServer ~]# chown root.root /etc/services
[root@myServer ~]# chmod 0644 /etc/services
[root@myServer ~]# chattr +i /etc/services
```

Ciò impedisce di rinominare, eliminare o di creare collegamenti al file.

2.8.2.4.1. Protocollo TCP

Queste sono le opzioni disponibili per il protocollo TCP (**`-p tcp`**):

- **`--dport`** — Specifica il numero di porta di destinazione.

Per configurare questa opzione, usare un nome (come www o smtp), un numero o un range di numeri di porta.

Per specificare un range di numeri, separare i due numeri con il carattere "due punti" (**:**). Per esempio: **-p tcp --dport 3000:3200**. Il range di valori massimo è **0:65535**.

Usare il carattere "punto esclamativo" (**!**) dopo l'opzione **--dport** per indicare i pacchetti che *non* usano quel servizio di rete o numero di porta.

Per conoscere i nomi e gli aliases dei servizi di rete con i numeri di porta usati, vedere il file **/etc/services**.

L'opzione **--destination-port** è la versione estesa di **--dport**.

- **--sport** — Specifica la porta mittente usando le stesse opzioni di **--dport**. L'opzione **--source-port** è la versione estesa di **--sport**.
- **--syn** — Applica la regola a tutti i pacchetti TCP designati ad iniziare la comunicazione, generalmente detti *SYN packet*. I pacchetti che trasportano dati (data payload) non ne sono influenzati.

  Usare il carattere "punto esclamativo" (**!**) dopo l'opzione **--syn** per indicare i pacchetti *non-SYN*.
- **--tcp-flags <tested flag list> <set flag list>** — Si applica ai pacchetti TCP che hanno impostati particolari bit (flag).

  L'opzione **--tcp-flags** accetta due parametri. Il primo è la maschera, una lista di flag separati da virgole da esaminare nel pacchetto. Il secondo parametro è una lista di flag separati da virgole che devono risultare settati.

  I flag possibili sono:

  - **ACK**
  - **FIN**
  - **PSH**
  - **RST**
  - **SYN**
  - **URG**
  - **ALL**
  - **NONE**

  Per esempio, la seguente regola si applica ai pacchetti TCP che hanno il flag SYN settato e i flag ACK e FIN non settato:

  --tcp-flags ACK,FIN,SYN SYN

  Usare il carattere punto esclamativo (**!**) dopo l'opzione**--tcp-flags** per invertire l'effetto della regola.
- **--tcp-option** — Applica la regola se è impostata l'opzione tcp. La regola può anche essere invertita usando il punto esclamativo (**!**).

2.8.2.4.2. Protocollo UDP

Queste sono le match option disponibili per il protocollo UDP (**`-p udp`**):

- **`--dport`** — Specifica la porta di destinazione usando il nome del servizio, il numero o un range di numeri di porta. L'opzione **`--destination-port`** è la versione estesa di **`--dport`**.
- **`--sport`** — Specifica la porta mittente usando il nome del servizio, il numero o un range di numeri di porta. L'opzione **`--source-port`** è la versione estesa di **`--sport`**.

Usando **`--dport`** e **`--sport`** per specificare un range di numeri, separare i due numeri con il carattere "due punti" (**`:`**). Per esempio: **`-p udp --dport 3000:3200`**. Il range di valori massimo è **`0:65535`**.

2.8.2.4.3. Protocollo ICMP

Per il protocollo ICMP (Internet Control Message Protocol) (**`-p icmp`**) sono disponbili le seguenti match option:

- **`--icmp-type`** — Specifica il nome o il numero del tipo di ICMP. Per la lista dei nomi di ICMP validi usare il comando **`iptables -p icmp -h`**.

2.8.2.4.4. Ulteriori moduli Match Option

Altre match option sono disponibili nei moduli caricati dal comando **`iptables`**.

Per usare un modulo, caricare il modulo per nome usando l'opzione **`-m`** ***<nome-del-modulo>***.

Per impostazione sono disponibili molti moduli. Si possono anche creare moduli personalizzati per aggiungere ulteriori funzionalità.

Di seguito si riporta un elenco (parziale) dei moduli maggiormente usati:

- modulo **`limit`** — Specifica quante volte applicare la regola.

  Assieme al "target" **`LOG`**, il modulo **`limit`** serve ad impedire che un flusso consistente di pacchetti possa riempire il file di log con messaggi ripetitivi o ad impedire di sovraccaricare il sistema.

  Per maggiori informazioni sul target **`LOG`**, fare riferimento alla *Sezione 2.8.2.5, «Opzioni target»*.

  Il modulo **`limit`** presenta le seguenti opzioni:

  - **`--limit`** — Imposta il numero massimo di corrispondenze per periodo, usando la coppia ***<value>/<period>***. Per esempio, specificando **`--limit 5/hour`** si permettono cinque corrispondenze all'ora.

    Gli intervalli possono essere espressi in secondi, minuti, ore o giorni.

    Se non è specificato un numero o una stringa temporale si assume il valore predefinito **`3/hour`**.

  - **`--limit-burst`** — Imposta il limite sul numero di pacchetti contemporanei gestiti dalla regola.

    Questa opzione è specificata con un intero e dovrebbe essere usata insieme all'opzione **`--limit`**.

    Se non è specificato nessun valore, il valore predefinito è cinque (5).

- modulo **`state`** — Identifica lo stato di un pacchetto.

  Il modulo **`state`** presenta le seguenti opzioni:

- **`--state`** — Identifica un pacchetto con uno dei seguenti stati di connessione:
  - **`ESTABLISHED`** — Il pacchetto fa parte di una connessione già instaurata. Questo stato è indispensabile per il mantenimento della connessione tra client e server.
  - **`INVALID`** — Il pacchetto non fa parte di una connessione nota.
  - **`NEW`** — Il pacchetto tenta di creare una nuova connessione o fa parte di una connessione bidirezionale non ancora vista. Questo stato è indispensabile per creare connessioni.
  - **`RELATED`** — Il pacchetto tenta di avviare una nuova connessione, legata in qualche mdo ad una connessione già esistente. Un esempio è il protocollo FTP che usa una connessione sulla porta 21 per il controllo del traffico ed una connessione separata sulla porta 20 per il trasferimento dei dati.

  Questi stati di connessione possono essere usati in combinazione, separandoli con virgole come in **`-m state --state INVALID,NEW`**.

- modulo **`mac`** — Identifica l'indirizzo hardware MAC.

  Il modulo **`mac`** presenta la seguente opzione:

  - **`--mac-source`** — Identifica i pacchetti spediti dall'indirizzo MAC della scheda di rete. Per escludere un indirizzo da una regola, usare il carattere punto esclamativo (**`!`**) dopo l'opzione **`--mac-source`**.

Per altre opzioni disponibili con i moduli, fare riferimento alle pagine di man di **`iptables`**.

2.8.2.5. Opzioni target

Quando un pacchetto viene intercettato da una regola, il pacchetto può essere inviato a vari "target" che intraprendono l'azione appropriata. Ogni catena hanno un target predefinito che entra in azione se nessuna regola nella catena è in grado di intercettare il pacchetto o se la regola corrispondente è priva di un target specifico.

Di seguito si riportano i target standard:

- ***`<user-defined-chain>`*** — Una catena definita dall'utente. In nomi della catene devono essere unici. Il target passa il pacchetto alla catena specificata.
- **`ACCEPT`** — Invia il pacchetto alla sua destinazione o ad un'altra catena.
- **`DROP`** — Scarta il pacchetto senza rispondere. Il sistema che ha spedito il pacchetto non viene avvisato dell'insuccesso.
- **`QUEUE`** — Il pacchetto è messo in coda per essere gestito dall'applicazione dello spazio utente.
- **`RETURN`** — Interrompe il controllo delle regole sul pacchetto. Se il pacchetto viene intercettato in una catena interna alla principale, il pacchetto è restituito alla catena principale da cui vengono riavviate le verifiche rimaste in sospeso. Se il target **`RETURN`** viene usato in una catena predefinita e il pacchetto non può ritornare alla catena precedente, per la catena corrente si usa il target predefinito.

In aggiunta sono disponibili estensioni con cui definire altri target, detti moduli "target" o moduli "match option", tuttavia la maggior parte si applicano soltanto a particolari tabelle e situazioni. Per maggiori informazioni sui moduli "match option", fare riferimento alla *Sezione 2.8.2.4.4, «Ulteriori moduli Match Option»*.

Esistono molti moduli target, la maggior parte dei quali si applicano a tabelle e situazioni specifiche. Alcuni dei moduli più comuni inclusi per impostazione in Fedora, sono:

- **`LOG`** — Registra nel file di log tutti i pacchetti intercettati dalla regola. Poichè i pacchetti sono individuati dal kernel, è il file **`/etc/syslog.conf`** che determina in quale file registrare questi avvisi (logs). Per impostazione, i logs si trovano nel file **`/var/log/messages`**.

  Le opzioni che si possono usare con il target **`LOG`** sono:

  - **`--log-level`** — Imposta il livello di priorità degli eventi di log. Per una lista dei livelli di priorità, fare riferimento alle pagine di man di **`syslog.conf`**.
  - **`--log-ip-options`** — Registra tutte le opzioni impostate nell'header di un pacchetto IP.
  - **`--log-prefix`** — Antepone una stringa di caratteri (max. 29) davanti ad ogni riga di log. Ciò può essere molto utile in fase di analisi dei pacchetti per realizzare filtri di syslog.

    Nota

    A causa di un problema potrebbe essere necessario inserire uno spazio davanti al valore del parametro *`log-prefix`*.

  - **`--log-tcp-options`** — Registra tutte le opzioni impostate nell'header di un pacchetto TCP.
  - **`--log-tcp-sequence`** — Registra la sequenza numerica TCP del pacchetto.

- **`REJECT`** — Scarta il pacchetto e restituisce al sistema remoto un pacchetto d'errore.

  Il target **`REJECT`** accetta l'opzione **`--reject-with <type>`** (in cui <*type*> è il tipo di rifiuto), permettendo di restituire insieme al pacchetto d'errore informazioni più dettagliate. Il messaggio **`port-unreachable`** è il tipo predefinito di errore. Per la lista completa di opzioni **<*type*>**, fare riferimento alle pagine di man di **`iptables`**.

Altri moduli target, tra cui alcuni molto utili per il mascheramento IP con la tabella **`nat`** o per l'alterazione dei pacchetti con la tabella **`mangle`**, possono trovarsi nelle pagine di man di **`iptables`**.

2.8.2.6. Elencare le opzioni

Il comando predefinito **`iptables -L [<chain-name>]`**, mostra le attuali catene nella tabella predefinita. Altre opzioni forniscono maggiori informazioni:

- **`-v`** — Visualizza un output più prolisso, per esempio il numero di pacchetti e byte analizzati da ogni catena, il numero di pacchetti e byte individuati da ogni regola e le schede di rete interessate da una particolare regola.
- **`-x`** — Espande i numeri al loro valore esatto. Il numero di pacchetti e bytes analizzati da una catena o regola risultano abbreviati in **`Kilobytes`**, **`Megabytes`** o **`Gigabytes`**. Questa opzione visualizza il valore esatto di pacchetti e byte.
- **`-n`** — Visualizza gli indirizzi IP e i numeri di porta in formato numerico, invece del formato predefinito basato su hostname e nome del servizio.

- **`--line-numbers`** — Elenca il numero d'ordine delle regole nella catena. Questa opzione risulta molto utile quando si vuole rimuovere una regola o per localizzare la posizione nelle catena in cui inserire una regola.
- **`-t <table-name>`** — Specifica un nome di tabella. Se omesso si fa riferiemento alla tabella predefinita.

2.8.3. Salvataggio delle regole IPTables

Le regole create con il comando **`iptables`** sono conservate in memoria. Se il sistema viene riavviato, prima del loro salvataggio, le regole **`iptables`** vengono perse. Per rendere persistenti al riavvio del sistema, le regole di filtraggio dei pacchetti (netfilter) esse devono essere salvate: come root, lanciare il comando:

```
/sbin/service iptables save
```

Il comando esegue lo script di init di **`iptables`** che a sua volta esegue il programma **`/sbin/iptables-save`**, scrivendo la configurazione di **`iptables`** corrente nel file **`/etc/sysconfig/iptables`**. Il file **`/etc/sysconfig/iptables`** esistente è salvato come **`/etc/sysconfig/iptables.save`**.

Al successivo riavvio del sistema, lo script di init di **`iptables`** ri-applica le regole salvate in **`/etc/sysconfig/iptables`** usando il comando **`/sbin/iptables-restore`**.

Normalmente, è sempre una buona norma testare una nuova regola di **`iptables`** prima di trasferirla nel file **`/etc/sysconfig/iptables`**; inoltre è possibile copiare le regole di **`iptables`** da un file di un altro sistema. Ciò permette una rapida distribuzione delle regole di **`iptables`** su più macchine.

Le regole possono essere salvate anche in un file separato per distribuzione, backup o altro. Per salvare le regole iptables, eseguire come root il seguente comando:

```
[root@myServer ~]# iptables-save > <filename> dove <filename> è il nome dato al gruppo di regole.
```

Importante

Se si distribuisce il file **`/etc/sysconfig/iptables`** su altre macchine, per renderle effettive, riavviare il servizio iptables digitando il comando **`/sbin/service iptables restart`**.

Nota

Notare la differenza tra il comando **`iptables`** *command* (**`/sbin/iptables`**), usato per manipolare tabelle e le relative catene, ed il comando **`iptables`** *service* (**`/sbin/iptables service`**), usato per abilitare e disabilitare il servizio **`iptables`** stesso.

2.8.4. Script di controllo IPTables

In Fedora, esistono due metodi di base per controllare **`iptables`**:

- **Amministrazione Firewall** (**`system-config-securitylevel`**) — Un'interfaccia grafica per creare, attivare e salvare le regole di un firewall di base. (Vedere la *Sezione 2.7.2, «Configurazione di un firewall di base»*).

- **`/sbin/service iptables <option>`** — Usato per manipolare varie funzionalità di **`iptables`** tramite i suoi script di init. Le opzioni disponibili sono:

  - **`start`** — Se è stato configurato un firewall (ossia, esiste il file **`/etc/sysconfig/iptables`**), tutte le istanze di **`iptables`** in esecuzione vengono arrestate e successivamente riavviate con il comando **`/sbin/iptables-restore`**. Questa opzione funziona solo se non è caricato il modulo del kernel, **`ipchains`**. Per verificare se il modulo è caricato, digitare come root il seguente comando:

    ```
    [root@MyServer ~]# lsmod | grep ipchains
    ```

    Se il comando non restituisce nessun output, vuol dire che il modulo non è stato caricato. In caso contrario usare il comando **`/sbin/rmmod`** per rimuovere il modulo.

  - **`stop`** — Se è in esecuzione un firewall, le regole di firewall in memoria sono scaricate insieme a tutti i moduli e ai componenti di iptables.

    Se nel file di configurazione **`/etc/sysconfig/iptables-config`** è stato modificato il valore della direttiva **`IPTABLES_SAVE_ON_STOP`** dal valore predefinito (no) al valore **`yes`**, le attuali regole sono salvate nel file **`/etc/sysconfig/iptables`** e le precedenti regole vengono salvate nel file **`/etc/sysconfig/iptables.save`**.

    Per maggiori informazioni, vedere la *Sezione 2.8.4.1, «File di configurazione degli script di controllo»*.

  - **`restart`** — Se è in esecuzione un firewall, le regole di firewall in memoria sono scaricate e il firewall è riavviato con le configurazioni presenti in **`/etc/sysconfig/iptables`**. Questa opzione funziona solo se il modulo del kernel **`ipchains`** non è caricato.

    Se nel file di configurazione **`/etc/sysconfig/iptables-config`** è stato modificato il valore della direttiva **`IPTABLES_SAVE_ON_RESTART`**, dal valore predefinito (no) al valore **`yes`**, le attuali regole sono salvate nel file **`/etc/sysconfig/iptables`** e le precedenti regole vengono salvate nel file **`/etc/sysconfig/iptables.save`**.

    Per maggiori informazioni, vedere la *Sezione 2.8.4.1, «File di configurazione degli script di controllo»*.

  - **`status`** — Visualizza lo stato del firewall ed elenca tutte le regole attive.

    La configurazione predefinita per questa opzione è visualizzare gli indirizzi IP in formato numerico. Per la visualizzazione in formato nome dominio ed hostname, impostare nel file **`/etc/sysconfig/iptables-config`** il valore della direttiva **`IPTABLES_STATUS_NUMERIC`** con il valore **`no`**. Per maggiori informazioni sul file di configurazione **`iptables-config`**, vedere la *Sezione 2.8.4.1, «File di configurazione degli script di controllo»*.

  - **`panic`** — Scarica tutte le regole di firewall. La policy di tutte le tabellle configurate viene impostata a **`DROP`**.

    Questa opzione potrebbe essere utile quando si scopre che un server è compromesso. Piuttosto che spegnere o fisicamente diconnettere il sistema dalla rete si può usare questa opzione per fermare ogni traffico da/verso la rete, portando la macchina in uno stato ideale per analisi o altre investigazioni.

- **save** — Salva le regole di firewall nel file **/etc/sysconfig/iptables** con il comando **iptables-save**. Per maggiori informazioni, vedere la *Sezione 2.8.3, «Salvataggio delle regole IPTables»*.

Nota

In IPv6 il controllo di netfilter avviene allo stesso modo come fin quì indicato, basta sostituire **ip6tables** con **iptables** nei comandi di **/sbin/service**. Per maggiori informazioni su IPv6 e netfilter, vedere *Sezione 2.8.5, «IPTables ed IPv6»*.

2.8.4.1. File di configurazione degli script di controllo

Il comportamento degli init-script di **iptables** è controllato dal file di configurazione **/etc/sysconfig/iptables-config**. Di seguto si riporta un elenco delle direttive contenute in questo file:

- **IPTABLES_MODULES** — All'avvio del firewall specifica una lista di moduli di **iptables** da caricare. Questi possono includere componenti NAT e tracciatori di connessione.
- **IPTABLES_MODULES_UNLOAD** — Al riavvio o all'arresto del firewall tutti i moduli vengono scaricati. Questa direttiva accetta i seguenti valori:
  - **yes** — Il valore predefinito. Usare questo valore al fine di garantire un corretto stato dopo un riavvio o arresto del firewall.
  - **no** — Usare questo valore soltanto se ci sono problemi nello scaricare i moduli.
- **IPTABLES_SAVE_ON_STOP** — All'arresto del firewall le regole correnti del firewall sono salvate nel file **/etc/sysconfig/iptables**. Questa direttiva accetta i seguenti valori:
  - **yes** — All'arresto del firewall le regole esistenti sono salvate nel file **/etc/sysconfig/iptables** e le regole precedenti sono spostate nel file **/etc/sysconfig/iptables.save**.
  - **no** — Il valore predefinito. All'arresto del firewall le regole esistenti vengono perse.
- **IPTABLES_SAVE_ON_RESTART** — Al riavvio del firewall le regole correnti vengono salvate. Questa direttiva accetta i seguenti valori:
  - **yes** — Al riavvio del firewall le regole esistenti sono salvate nel file **/etc/sysconfig/iptables** e le regole precedenti vengono salvare nel file **/etc/sysconfig/iptables.save**.
  - **no** — Il valore predefinito. Al riavvio del firewall le regole esistenti vengono perse.
- **IPTABLES_SAVE_COUNTER** — Salva e ripristina i contatori di pacchetti e byte nelle regole di tutte le catene. Questa direttiva accetta i seguenti valori:
  - **yes** — Salva i valori dei contatori.
  - **no** — Valore predefinito. I valori dei contatori vengono azzerati.
- **IPTABLES_STATUS_NUMERIC** — Visualizza gli indirizzi IP in formato numerico invece del formato basato su nomi (dominio ed hostname). Questa direttiva accetta due valori:
  - **yes** — Il valore predefinito. Restituisce gli indirizzi IP in formato numerico.

- **no** — Restituisce gli indirizzi in formato nome dominio ed hostname.

2.8.5. IPTables ed IPv6

Se è installato il pacchetto **`iptables-ipv6`** allora è possibilie filtrare i pacchetti (netfilter) del protocollo Internet IPv6 di prossima generazione. Il comando usato per manipolare il netfilter IPv6 è **`ip6tables`**.

Le principali direttive di questo comando sono identiche a quelle del comando **`iptables`**, ad eccezione della tabella **`nat`** non ancora supportata. Ciò vuol dire che ad oggi non è possibile effettuare operazioni NAT (Network Address Translation), sugli indirizzi IPv6 come il mascheramento e il forwarding dei servizi.

Le regole di **`ip6tables`** sono salvate nel file **`/etc/sysconfig/ip6tables`** e le regole precedenti vengono salvate nel file **`/etc/sysconfig/ip6tables.save`**.

Le opzioni di configurazione degli init-script si trovano nel file **`/etc/sysconfig/ip6tables-config`** e i nomi delle varie direttive variano di poco rispetto alle analoghe di **`iptables`**.

Per esempio, la direttiva **`IPTABLES_MODULES`** del file **`iptables-config`** è equivalente alla direttiva **`IP6TABLES_MODULES`** del file **`ip6tables-config`**.

2.8.6. Ulteriori risorse

Per altre informazioni sul filtraggio dei pacchetti con **`iptables`** fare riferimneto alle seguenti risorse.

- *Sezione 2.7, «Firewall»* — E' un capitolo dedicato al ruolo dei firewall nell'ambito di una strategia di sicurezza globale con strategie per costruire regole di firewall.

2.8.6.1. Documentazione installata

- **`man iptables`** — Contiene una descrizione di **`iptables`** con l'elenco completo dei targets, delle options e delle match extensions.

2.8.6.2. Utili siti web su IPTables

- *netfilter.org*[23] — Il sito web del progetto netfilter/iptables. Contiene informazioni assortite su **`iptables`**, inclusa una FAQ con soluzioni per problemi specifici e varie guide scritte da Rusty Russell, il manutentore del firewall IP di Linux. Gli HOWTO, coprono vari argomenti come concetti di rete, filtraggio dei pacchetti nel kernel e configurazioni NAT.

- *justlinux.com*[24] — Un'introduzione su come i pacchetti attraversano lo stack di comunicazione nel kernel Linux, con un'introduzione su come costruire comandi **`iptables`** di base.

---

[23] http://www.netfilter.org/
[24] http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html

Cifratura

Esistono due principali tipi di dati che devono essere protetti: i dati a riposo e i dati in movimento. Questi differenti tipi di dati sono protetti in modo simile, usando tecnologie simili ma le implementazioni possono essere completamente differenti. Nessuna implementazione, per quanto sicura, può sentirsi tale contro tutti i possibili metodi di compromissione, proprio perchè l'informazione può essere a riposo e in movimento in differenti istanti di tempo.

3.1. Dati a Riposo

I dati a riposo sono i dati immagazzinati su disco fisso, nastro, CD, DVD o altro supporto. La principale minaccia contro questo tipo di dati è rappesentata dal furto. I portatili negli aereoporti, i CD spediti per posta e i nastri di backup che vengono lasciati nei posti sbagliati sono tutti esempi di eventi in cui i dati possono essere compromessi da un furto. Se i dati sono stati cifrati allora non c'è da preoccuparsi così tanto della loro compromissione.

3.1.1. Completa cifratura del disco

La completa cifratura del disco o di una sua partizione, rappresenta uno dei metodi migliori per proteggere i dati. Non solo è protetto ogni file ma anche la memoria temporanea contenente parti di questi file. La completa cifratura del disco è in grado di proteggere tutti i file, evitando all'utente la preoccupazione di quali file proteggere ed eventuali sue dimenticanze.

Fedora 14 (e le versioni precedenti fino a Fedora 9), supporta in modo nativo la cifratura LUKS. LUKS cifra le partizioni del disco fisso proteggendo i dati quando il computer è inattivo. Inoltre protegge il computer anche da attaccanti che in modalità *single user* o in altro modo riescono ad accedere al computer.

Soluzioni di cifratura del disco come LUKS, proteggono i dati solo quando il computer è spento. Una volta attivo e decifrato da LUKS, i file sul disco diventano disponibili a chiunque abbia accesso alla macchina. Per proteggere i file quando il computer è acceso, usare la cifratura del disco in combinazione con un'altra soluzione, come la cifratura basata su file. Ricordare inoltre che è buona norma bloccare il computer, ogni qualvolta ci si allontana dalla propria postazione. Impostare un salvaschermo protetto da frase d'accesso che si attivi dopo qualche minuto di inattività, è un buon modo per mantenere lontani eventuali intrusi.

3.1.2. Cifratura basata su file

GnuPG (GPG) è una versione open source di PGP che consente di firmare e/o cifrare un file o un messaggio email. Ciò serve a garantire l'integrità del messaggio o del file ed inoltre protegge la confidenzialità delle informazioni contenute. Nel caso delle mail GPG fornisce una doppia protezione. Non solo fornisce la protezione dei Dati a Riposo ma anche dei Dati in Movimento.

La cifratura basata su file serve a proteggere il file dopo che esso ha lasciato il computer, come quando si spedisce un CD per posta. Alcune soluzioni lasciano dei residui del file cifrato, che un attaccante con accesso fisico al computer, in determinate circostanze, può usare per ripristinare il file cifrato. Per proteggere i contenuti di questi file da utenti maliziosi, usare la cifratura basata su file in combinazione con altre soluzioni, come la completa cifratura del disco.

3.2. Dati in Movimento

I dati in movimento sono dati che vengono trasmessi nella rete. Le principali minacce contro i dati in movimento sono l'intercettazione e l'alterazione. Password e Nome Utente non dovrebbero

essere mai trasmessi nella rete senza protezione, poichè potrebbero essere intercettate e usate da qualcun'altro per impersonare l'utente e/o per guadagnare l'accesso ad informazioni sensibili. Anche altre informazioni private, come quelle relative ai conti bancari, dovrebbero essere protette quando vengono trasmesse in una rete. Se la sessione di rete è stata cifrata allora non si corre alcun rischio: i dati non possono venir compromessi durante la trasmissione.

I dati in movimento sono particolarmente vulnerabili agli attaccanti, in quanto questi non devono trovarsi nei pressi della postazione del computer, dove sono salvati i dati, ma possono trovarsi ovunque lungo il percorso seguito dai dati. Tunnel di cifratura possono proteggere i dati lungo il percorso di comunicazione.

3.2.1. Virtual Private Networks (VPN)

Le organizzazioni con uffici dislocati in diverse località, per motivi di efficenza e proteggere i dati sensibili, spesso sono connessi tramite linee dedicate. Per esempio, molte attività commerciali usano linee frame relay o *ATM* (Asynchronous Transfer Mode), come soluzioni di rete end-to-end per il collegamento degli uffici. Tuttavia per le piccole e medie imprese (n.d.t.: e l'Italia fonda il suo PIL sull'attività di circa l'80% di tali imprese!) che desiderano espandersi, investire in tale soluzioni, richiede alti costi di investimento in circuiti di rete digitali, molte volte ben al di là dei propri bilanci aziendali.

Le reti *VPN* (Virtual Private Networks) sono state progettate proprio per venire incontro a queste esigenze aziendali. Seguendo gli stessi principi funzionali dei circuiti dedicati, le reti VPN consentono comunicazioni digitali sicure tra due partecipanti (o reti), creando una *WAN* (Wide Area Network) a partire da *LAN* (Local Area Network) esistenti. La differenza rispetto a linee frame relay o ATM è il mezzo di trasporto. Le reti VPN trasportano i dati sul layer IP, usando pacchetti, attraverso un canale sicuro che attraverso Internet giunge alla rete di destinazione. Le principali implementazioni free di VPN, incorporano metodi di cifratura standard ed aperti, per ulteriormente mascherare i dati in transito.

Alcune organizzazioni impiegano soluzioni VPN hardware per aumentare la sicurezza, altre usano implementazioni software o basate su protocollo. Esistono diversi produttori di soluzioni VPN hardware, come Cisco, Nortel, IBM e Checkpoint. Esiste una soluzioneVPN basata su software free anche per Linux, denominata FreeS/Wan, che utilizza una implementazione standardizzata di IPsec (*Internet Protocol Security*). Le soluzioni VPN sia hardware sia software, si comportano come router specializzati tra le connessioni IP dei vari uffici.

3.2.1.1. Come funziona una rete VPN?

Quando un pacchetto viene trasmesso da un client, esso passa attraverso il router o gateway del VPN, che aggiunge un AH (*Authentication Header*) usato per routing ed autenticazione. Successivamente i dati vengono cifrati e poi racchiusi in un ESP (*Encapsulating Security Payload*). All'interno di quest'ultimo si trovano le istruzioni per gestire e decifrare il pacchetto.

Il router del VPN ricevente, estrae le informazioni dall'intestazione, decifra i dati e invia i dati alla sua destinazione (una workstation o un altro nodo della rete). In una connessione network-to-network, il nodo ricevente sulla rete locale, riceve i pacchetti già decifrati e pronti per l'uso. Il processo di cifratura/decifratura in una connessione VPN network-to-network, è quindi trasparente al nodo locale.

Con un tale livello di sicurezza, un attaccante non solo deve intercettare il pacchetto, ma anche decifrarlo. Intrusori che impiegano un attacco tipo man-in-the-middle, devono avere accesso anche ad almeno una chiave segreta per l'autenticazione delle sessioni. Poichè queste usano diversi livelli di autenticazione e di cifratura, le reti VPN sono un mezzo sicuro ed efficace per collegare multipli nodi remoti, che diventano così una intranet unificata.

3.2.1.2. Le reti VPN e Fedora

Fedora offre varie soluzioni per implementare una connessione sicura ad una WAN. IPsec (*Internet Protocol Security*) è l'implementazione VPN supportata in Fedora, in grado di soddisfare adeguatamente i bisogni di usabilità delle organizzazioni con uffici ramificati o utenti remoti.

3.2.1.3. IPsec

Fedora supporta IPsec per collegare tra loro reti ed host remoti, tramite un tunnel sicuro attraverso una rete pubblica come Internet. IPsec può essere implementato sia per una configurazione host-to-host (tra due workstation) sia per una configurazione network-to-network (tra due LAN/WAN).

L'implementazione di IPsec in Fedora usa *IKE* (*Internet Key Exchange*), un protocollo progettato dall'IETF (Internet Engineering Task Force) ed usato per reciproca autenticazione e associazioni sicure tra i sistemi.

3.2.1.4. Creare una connessione IPsec

Una connessione IPsec prevede due fasi logiche. Nella prima fase, un nodo IPsec inizializza la connessione con la rete o il nodo remoto. La rete o il nodo remoto controlla le credenziali del nodo richiedente, dopodichè entrambi i nodi negoziano il metodo di autenticazione da usare per la connessione.

Nei sistemi Fedora, una connessione di IPsec usa il metodo della *pre-shared key* (o della chiave pre-condivisa) per l'autenticazione dei nodi IPsec. In una connessione IPsec con chiave pre-condivisa, entrambi gli host devono usare la stessa chiave per poter passare alla seconda fase della connessione IPsec.

La seconda fase della connessione IPsec, prevede la creazione di una SA (*Security Association*) tra i nodi IPsec. Questa fase genera un database SA contenente informazioni di configurazioni, come il metodo di cifratura, parametri per lo scambio delle chiavi segrete ed altro. Questa fase gestisce l'effettiva connessione IPsec tra i nodi remoti o le reti.

L'implementazione di IPsec in Fedora, usa IKE per lo scambio, attraverso Internet, delle chiavi tra gli host. Il demone delle chiavi, **`racoon`** è addetto alla distribuzione e allo scambio della chiave IKE. Per maggiori informazioni su questo demone, vedere le pagine di man su **`racoon`**.

3.2.1.5. Installazione di IPsec

L'implementazione di IPsec richiede che il pacchetto **`ipsec-tools`** sia installato su tutti gli host IPsec (nel caso di una configurazione host-to-host) o router (nel caso di una configurazione network-to network). Il pacchetto contiene le librerie, i demoni e i file di configurazione essenziali per impostare una connessione IPsec, inclusi:

- **`/sbin/setkey`** — regola il gestore delle chiavi e gli attributi di sicurezza di IPsec nel kernel. Questo eseguibile è controllato dal processo **`racoon`**, il demone gestore delle chiavi. Per i dettagli, vedere le pagine di man su **`setkey`**(8).
- **`/usr/sbin/racoon`** — il demone che gestisce le chiavi IKE, usato per gestire e controllare la sicurezza delle associazioni e lo scambio delle chiavi tra i sistemi IPsec.
- **`/etc/racoon/racoon.conf`** — il file di configurazione del demone **`racoon`**, usato per impostare vari aspetti di una connessione IPsec, inclusi i metodi di autenticazione e gli algoritmi di cifratura da usare nella connessione. Per una lista completa delle direttive disponibili, vedere le pagine di man relative a **`racoon.conf`**(5).

Per configurare IPsec su un sistema Fedora, si può usare l'interfaccia grafica di **Amministrazione della rete**, o procedere manualmente modificando i file di configurazione di rete e di IPsec.

- Per connettere tra loro via IPsec, due host di una rete, vedere *Sezione 3.2.1.6, «Configurazione IPSec Host-to-Host»*.
- Per connettere tra loro via IPsec, due LAN/WAN, vedere *Sezione 3.2.1.7, «Configurazione IPsec Network-to-Network»*.

3.2.1.6. Configurazione IPSec Host-to-Host

IPsec può essere configurato per collegare tra loro due desktop o workstation (host), usando una connessione host-to host. Questo tipo di connessione usa la rete a cui è connesso ciascun host, per creare un tunnel sicuro tra i due host. Le specifiche richieste per creare una connessione host-to-host sono minime, come risulta la configurazione di IPsec su ciascun host. Gli host necessitano solo di una connessione alla rete portante (come Internet) e di un sistema Fedora per creare la connessione IPsec.

3.2.1.6.1. Connessione Host-to-Host

Una connessione IPsec Host-to-Host, è una connessione cifrata tra due sistemi, in quanto su entrambi gli host, IPsec usa la stessa chiave di autenticazione. Con la connessione IPsec attiva, tutto il traffico di rete tra i due host risulta cifrato.

Per configurare una connessione IPsec host-to-host, procedere su ciascun host, come indicato:

Nota

Le seguenti procedure dovrebbero essere eseguite direttamente sulla macchina: si raccomanda di evitare configurazioni e connessioni IPsec da remoto.

1. In un terminale, digitare **`system-config-network`** per avviare l'interfaccia grafica di **Amministrazione della rete**, oppure dal menu d'avvio selezionare **Sistema > Amministrazione > Amministrazione della rete**.
2. Nella scheda **IPsec**, premere sul pulsante **Nuovo** per avviare il wizard di configurazione.
3. Premere **Avanti** per avviare la configurazione di una connessione IPsec host-to-host.
4. Inserire un nome unico da assegnare alla connessione, per esempio **`ipsec0`**. Se si desidera attivare la connessione automaticamente, all'avvio del computer, spuntare la casella di controllo. Premere **Avanti** per continuare.
5. Selezionare come tipo di connessione, **Crittografia da Host to Host** e poi premere **Avanti**.
6. Selezionare il tipo di cifratura da usare: manuale o automatica.

   Se si sceglie la cifratura manuale, successivamente occorrerà fornire una chiave di cifratura. Se si seleziona la cifratura automatica, sarà il demone **`racoon`** a creare la chiave di cifratura. Se si usa la cifratura automatica, occorre che sia installato il pacchetto **`ipsec-tools`**.

   Premere **Avanti** per continuare.
7. Inserire l'indirizzo IP dell'host remoto.

   Per determinare l'IP dell'host remoto, usare il seguente comando, *sull'host remoto*:

```
[root@myServer ~] # /sbin/ifconfig <device>
```

dove *<device>* è la scheda di rete (Ethernet) usata per la connessione VPN.

Se è presente una sola scheda di rete nel sistema, il dispositivo tipicamente è denominato eth0. Di seguito si riporta un esempio, con le informazioni rilevanti dell'output di questo comando:

```
eth0      Link encap:Ethernet  HWaddr 00:0C:6E:E8:98:1D
          inet addr:172.16.44.192  Bcast:172.16.45.255  Mask:255.255.254.0
```

L'indirizzo IP è dato dal numero appresso alla stringa **`inet addr:`**.

Nota

Per connessioni host-to-host, entrambi gli host devono possedere un indirizzo pubblico. Altrimenti, se si trovano sulla stessa LAN, possono avere un indirizzo privato (p.e. indirizzi nel range 10.x.x.x o 192.168.x.x).

Nel caso i due host si trovino su differenti LAN, oppure se un host ha un indirizzo pubblico e l'altro un indirizzo privato, vedere la *Sezione 3.2.1.7, «Configurazione IPsec Network-to-Network»*.

Premere **Avanti** per continuare.

8. Se al passo 6, è stata selezionata la cifratura manuale, specificare la chiave di cifratura da usare, oppure premere **Genera** per crearne una.

   a. Specificare una chiave di autenticazione o premere **Genera** per crearne una. Si può usare una qualsiasi combinazione di lettere e numeri.

   b. Premere **Avanti** per continuare.

9. Nella pagina **IPsec — Sommario**, rivedere le informazioni inserite e poi premere **Applica**.

10. Per salvare la configurazione creata, selezionare **File** => **Salva**.

    Per rendere effettive le modifiche potrebbe essere necessario riavviare la rete. In tal caso, usare il seguente comando:

```
[root@myServer ~]# service network restart
```

11. Dalla lista delle connessioni IPsec, selezionare la connessione appena creata e premere il pulsante **Attiva**.

12. Ripetere l'intera procedura sull'altro host, prestando particolare attenzione ad usare la stessa chiave usata nel passo 8, sul primo host. Pena il non funzionamento di IPsec.

Dopo aver configurato la connessione IPsec, essa compare nella scheda di IPsec come indicato in *Figura 3.1, «Connessione IPsec»*.

Figura 3.1. Connessione IPsec

Alla fine del processo di creazione della connessione IPsec, vengono generati i seguenti file:

- **`/etc/sysconfig/network-scripts/ifcfg-<nickname>`**
- **`/etc/sysconfig/network-scripts/keys-<nickname>`**
- **`/etc/racoon/<remote-ip>.conf`**
- **`/etc/racoon/psk.txt`**

Se è stata usata la cifratura automatica, verrà creato anche il file **`/etc/racoon/racoon.conf`**.

Quando la connessione è attiva, il file **`/etc/racoon/racoon.conf`** viene modificato per includere **`<remote-ip>.conf`**.

3.2.1.6.2. Configurazione manuale di IPsec Host-to-Host

Prima di procedere, recuperare le informazioni di sistema e di rete di ogni workstation. Per una connessione host-to-host, occorre conoscere:

- L'indirizzo IP degli host
- Un nome unico (p.e. **`ipsec1`**), identificativo della connessione IPsec. Serve ad identificare la connessione IPsec ed a distinguerla da altre connessioni.
- Una chiave di cifratura fissata o una generata automaticamente da **`racoon`**.
- Una chiave di autenticazione pre-condivisa, usata durante la fase iniziale della connessione e per lo scambio delle chiavi cifrate durante la sessione.

Per esempio, si supponga che la workstation A e la workstation B vogliano connettersi tra loro attraverso un tunnel IPsec. Essi vogliono connettersi usando un chiave pre-condivisa il cui valore è

Key_Value01, e decidono di usare **racoon** per generare automaticamente e condividere una chiave per l'autenticazione reciproca. Entrambi gli utenti decidono di chiamare **ipsec1** le loro connessioni.

Nota

Si consiglia di usare una chiave PSK con una combinazione di lettere maiuscole/minuscole, numeri e caratteri di punteggiatura. Una chiave PSK facile da scoprire costituisce un rischio alla sicurezza.

Non è necessario usare, sui due host, lo stesso nome per la connessione. Si potrebbe scegliere un nome che sia significativo per la propria installazione.

Di seguito si riporta il file di configurazione di IPsec della prima workstation A per una connessione IPsec host-to host con la workstation B. L'identificativo della connessione usato nell'esempio è *ipsec1*, per cui il file di configurazione è **/etc/sysconfig/network-scripts/ifcfg-ipsec1**:

```
DST=X.X.X.XTYPE=IPSEC
ONBOOT=no
IKE_METHOD=PSK
```

Per la workstation A, *X.X.X.X* è l'indirizzo IP della workstation B. Per la workstation B, *X.X.X.X* è l'indirizzo IP della workstation A. La connessione è configurata in modo da non avviarsi al boot di sistema (**ONBOOT=no**) ed usa il metodo di autenticazione della chiave pre-condivisa (**IKE_METHOD=PSK**).

Di seguito si mostra il contenuto del file della chiave pre-condivisa (denominato **/etc/sysconfig/network-scripts/keys-ipsec1**), usato da entrambe le workstation per autenticarsi tra loro. Il suo contenuto dovrebbe essere identico nelle due workstation, il cui accesso in lettura/scrittura, dovrebbe essere consentito solo all'utente root.

```
IKE_PSK=Key_Value01
```

Importante

Per modificare i permessi al file **keys-ipsec1** in modo che solo l'utente root possa leggere o modificare il file, usare il seguente comando:

```
[root@myServer ~] # chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

Per modificare la chiave di autenticazione, editare il file **keys-ipsec1** su entrambe le workstation. *Le chiavi di autenticazione devono coincidere perchè la connessione funzioni correttamente.*

Il successivo esempio, mostra la configurazione propria alla fase 1 della connessione con l'host remoto. Il file è denominato ***X.X.X.X*.conf**, in cui *X.X.X.X* è l'indirizzo IP dell'host IPsec remoto. Notare che questo file è generato automaticamente all'avvio del tunnel IPsec e non dovrebbe essere eplicitamente modificato.

```
remote X.X.X.X{
         exchange_mode aggressive, main;
  my_identifier address;
  proposal {
   encryption_algorithm 3des;
  hash_algorithm sha1;
  authentication_method pre_shared_key;
  dh_group 2 ;
 }
}
```

Il file di configurazione della fase 1 viene creato durante l'inizializzazione della connessione IPsec e nell'implementazione di IPsec di Fedora, contiene le seguenti istruzioni:

remote *X.X.X.X*
: Specifica che le seguenti istruzioni di questo file di configurazione, si applicano solo al nodo remoto identificato dall'indirizzo IP *X.X.X.X*.

exchange_mode aggressive
: La configurazione predefinita di IPsec in Fedora usa un metodo di autenticazione *aggressive*, che riduce lo scambio di informazioni di connessione per consentire di configurare più connessioni IPsec con host multipli.

my_identifier address
: Specifica il metodo di identificazione da usare per autenticare i nodi. Fedora usa indirizzi IP per identificare i nodi.

encryption_algorithm 3des
: Specifica l'algoritmo di cifratura da usare durante l'autenticazione. Per impostazione, si usa 3DES (*Triple Data Encryption Standard*).

hash_algorithm sha1;
: Specifica l'algoritmo di hash da usare durante la negoziazione della fase 1. Per impostazione, si usa SHA (Secure Hash Algorithm version 1).

authentication_method pre_shared_key
: Specifica il metodo di autenticazione da usare durante la negoziazione tra i nodi. Per impostazione, Fedora usa chiavi pre-condivise per l'autenticazione.

dh_group 2
: Specifica il numero di gruppo di Diffie-Hellman con cui avviare lo scambio delle chiavi. Per impostazione, si usa modp1024 (group 2).

3.2.1.6.2.1. Il file di configurazione di racoon

Il file **/etc/racoon/racoon.conf** dovrebbe essere identico in tutti i nodi IPsec, con l'*eccezione* dell'istruzione **include "/etc/racoon/*X.X.X.X*.conf"**. Per la workstation A, *X.X.X.X* nell'istruzione **include** rappresenta l'indirizzo IP della workstation B; mentre nel file della workstation B, rappresenta l'indirizzo IP della workstation A. Di seguito si riporta un file **racoon.conf** tipico, in una connessione IPsec attiva:

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
```

```
sainfo anonymous
{
        pfs_group 2;
        lifetime time 1 hour ;
        encryption_algorithm 3des, blowfish 448, rijndael ;
        authentication_algorithm hmac_sha1, hmac_md5 ;
        compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf";
```

Il file **`racoon.conf`** predefinito, include i percorsi relativi alla configurazione di IPsec, ai file della chiave pre-condivisa ed ai certificati d'autenticazione. I campi in **`sainfo anonymous`** descrivono una SA tra i nodi IPsec della fase 2 — la natura della connessione IPsec, il tipo di algoritmo di cifratura usato e il metodo di scambio delle chiavi. Di seguito si definiscono i campi della fase 2:

sainfo anonymous
: Denota che una SA può inizializzarsi in maniera anonima con ogni peer purchè coincidano le credenziali IPsec.

pfs_group 2
: Definisce il protocollo Diffie-Hellman per lo scambio chiavi, il metodo usato dai nodi IPsec per stabilire la chiave di comunicazione segreta per la seconda fase della connessione IPsec. Per impostazione, l'implementazione di IPsec in Fedora, usa il Group 2 (o **`modp1024`**) di Diffie-Hellman per lo scambio delle chiavi segrete. Group 2 usa chiavi generate in modulo a 1024-bit, per impedire ad attaccante eventualmente in possesso di chiavi compromesse, la decifrazione di precedenti trasmissioni IPsec.

lifetime time 1 hour
: Questo parametro specifica il tempo di vita medio di una SA e può essere espresso in formato orario o di data. Per impostazione, in Fedora si specifica in ore.

encryption_algorithm 3des, blowfish 448, rijndael
: Specifica l'algoritmo di cifratura della fase 2. Fedora supporta gli algoritmi 3DES, 448-bit Blowfish e Rijndael (l'algoritmo usato in AES o *Advanced Encryption Standard*).

authentication_algorithm hmac_sha1, hmac_md5
: Elenca gli algoritmi di hash supportati per l'autenticazione. Quelli supportati sono HMAC-SHA1 e HMAC-MD5.

compression_algorithm deflate
: Definisce l'algoritmo di compressione Deflate a supporto di IPCOMP (IP Payload Compression), per consentire trasmissioni di datagram IP più veloci, su connessioni lente.

Per avviare la connessione, su ciascun host usare il seguente comando:

```
[root@myServer ~]# /sbin/ifup <nickname>
```

in cui <nickname> è il nome della connessione IPsec.

Per testare la connessione IPsec, eseguire l'utility **`tcpdump`** che visualizza i pacchetti trasferiti tra gli host e verifica se sono cifrati via IPsec. Il pacchetto dovrebbe includere un'intestazione AH ed essere segnato come ESP, ad indicare che si tratta di un pacchetto cifrato. Per esempio:

```
[root@myServer ~]# tcpdump -n -i eth0 host <targetSystem>

IP 172.16.45.107 > 172.16.44.192: AH(spi=0x0954ccb6,seq=0xbb): ESP(spi=0x0c9f2164,seq=0xbb)
```

3.2.1.7. Configurazione IPsec Network-to-Network

IPsec può anche essere configurato per connettere una rete (come una LAN o WAN), ad una rete remota usando una connessione network-to-network. Una tale connessione richiede di impostare i router IPsec sulle due reti in maniera da processare e indirizzare con trasparenza, le informazioni in transito da un nodo della LAN a un nodo della LANremota. La *Figura 3.2, «Una connessione IPsec network-to-network»* illustra una tipica connessione IPsec network-to-network.

Figura 3.2. Una connessione IPsec network-to-network

Lo schema mostra due LAN separate da Internet. Le LAN usano router IPsec per autenticare e iniziare una connessione, usando un tunnel sicuro attraverso Internet. I pacchetti intercettati da malintenzionati, richiederebbero dei sistemi di decifrazione molto potenti, in quanto dovrebbero verificare iterativamente tutte le combinazioni di chiavi possibili (brute-force decryption). Il processo di comunicazione tra un nodo della rete 192.168.1.0/24 ed un altro della rete 192.168.2.0/24 risulta completamente trasparente agli altri nodi poichè la cifratura/decifratura e il routing dei pacchetti IPsec sono interamente gestiti dai router IPsec.

Le informazioni richieste per una connessione network-to-network, sono:

- Gli indirizzi IP esternamente accessibili dei router IPsec dedicati.
- Gli indirizzi di rete delle LAN/WAN servite dai router IPsec (per esempio 192.168.1.0/24 10.0.1.0/24)
- Gli indirizzi IP dei gateway che indirizzano i pacchetti dai nodi della rete verso Internet.
- Un nome unico (p.e. **`ipsec1`**), identificativo della connessione IPsec. Serve ad identificare la connessione IPsec ed a distinguerla da altre connessioni.
- Una chiave di cifratura fissata o una generata automaticamente da **`racoon`**
- Una chiave di autenticazione pre-condivisa, usata durante la fase iniziale della connessione e per lo scambio delle chiavi cifrate durante la sessione.

3.2.1.7.1. Connessione (VPN) Network-to-Network

Una connessione IPsec network-to-network usa due router IPsec, uno per ciascuna rete, attraverso cui passa il traffico diretto alle sotto-reti private.

Per esempio, come mostrato nella *Figura 3.3, «IPsec Network-to-Network»*, se la rete privata 192.168.1.0/24 invia dei pacchetti alla rete privata 192.168.2.0/24, i pacchetti passano dal gateway0 al nodo ipsec0, poi attraversano Internet e dal nodo ipsec1 al gateway1, arrivano alla rete 192.168.2.0/24.

I router IPsec richiedono due indirizzi IP pubblici ed una seconda scheda di rete connessa alla propria rete privata. Il traffico passa attraverso un router IPsec soltanto se è destinato al router IPsec con il quale ha una connessione cifrata.

Figura 3.3. IPsec Network-to-Network

Configurazioni alternative possono includere un firewall tra ciascun router IP e Internet, ed un firewall intranet tra ciascun router IPsec e il gateway della sotto-rete. Il router IPsec ed il gateway della sottorete possono anche coincidere con un unico sistema con due scede di rete: una con un IP pubblico che agisce da router IPsec; l'altra con un IP privato che agisce da gateway per la sottorete privata. Ciascun router IPsec può usare il gateway della propria rete o un gateway pubblico per trasmettere i pacchetti all'altro router IPsec.

Per configurare una connessione network-to-network IPsec, usare la seguente procedura:

1. In un terminale, digitare **`system-config-network`** per avviare l'interfaccia grafica di **Amministrazione della rete**, oppure dal menu d'avvio selezionare **Sistema > Amministrazione > Amministrazione della rete**.

2. Nella scheda **IPsec**, premere sul pulsante **Nuovo** per avviare il wizard di configurazione.

3. Premere **Avanti** per avviare la configurazione di una connessione IPsec network-to-network.

4. Inserire un nome unico con cui indicare la connessione, per esempio **`ipsec0`**. Se si desidera attivare automaticamente la connessione all'avvio del computer, attivare la casella di controllo. Premere **Avanti** per continuare.

5. Selezionare **Crittografia da rete a rete (VPN)**, per il tipo di connessione e poi premere **Avanti**.

6. Selezionare il tipo di cifratura da usare: manuale o automatica.

   Se si sceglie la cifratura manuale, successivamente occorrerà fornire una chiave di cifratura. Se si seleziona la cifratura automatica, sarà il demone **`racoon`** a creare la chiave di cifratura. Se si usa la cifratura automatica, occorre che sia installato il pacchetto **`ipsec-tools`**.

   Premere **Avanti** per continuare.

7. Nella scheda **Rete Locale**, inserire le seguenti informazioni:

   - **Indirizzo locale** — L'indirizzo IP della scheda di rete sul router IPsec connesso alla rete privata.

   - **Maschera di sottorete locale** — La subnet mask dell'indirizzo IP della rete locale

   - **Gateway della rete locale** — L'indirizzo del gateway per la sottorete privata

   Premere **Avanti** per continuare.

Figura 3.4. Informazioni di rete locale

8. Nella pagina **Rete remota**, inserire le seguenti informazioni:

   - **Indirizzo IP remoto** — L'indirizzo IP pubblico del router IPsec dell'*altra* rete privata. Nel nostro caso, per il router ipssec0, inserire l'IP del router ipsec1.

   - **Indirizzo di rete remota** — L'indirizzo della sottorete dietro all'*altro* router IPsec. Nel nostro esempio, inserire **`192.168.1.0`** se si configura ipsec1, e **`192.168.2.0`** se si configura ipsec0.

   - **Maschera di sottorete remota** — La maschera della sottorete remota.

   - **Gateway della rete remota** — L'indirizzo IP del gateway per la rete remota.

   - Se nel passo 6 si è scelta la cifratura maunale, specificare la chiave di cifratura da usare o premere **Genera** per crearne una.

     Specificare una chiave di autenticazione o premere **Genera** per crearne una. Questa chiave può essere una combinazione di numeri, lettere e caratteri di punteggiatura.

   Premere **Avanti** per continuare.

Figura 3.5. Informazioni di rete remota

9. Nella pagina **IPsec — Sommario**, rivedere le informazioni inserite e poi premere **Applica**.

10. Per salvare la configurazione, selezionare **File** => **Salva**.

11. Per attivare la connessione, selezionare la connessione IPsec dalla lista, e poi premere **Attiva**.

12. Abilitare l'IP forwarding:

    a. Modificare il file **`/etc/sysctl.conf`** impostando **`net.ipv4.ip_forward`** su **`1`**.

    b. Usare il seguente comando per rendere effettiva la modifica:

    ```
    [root@myServer ~]# /sbin/sysctl -p /etc/sysctl.conf
    ```

Lo script di rete che attiva automaticamente la connessione IPsec, crea i percorsi di instradamento dei pacchetti, trasmettendoli, se necessario, attraverso il router IPsec.

3.2.1.7.2. Configurazione manuale di IPsec Network-to-Network

Si supponga di voler connettere due LAN, A (lana.example.com) e B (lanb.example.com), usando un tunnel IPsec. La LAN A ha indirizzo 192.168.1.0/24, la LAN B 192.168.2.0/24. Gli indirizzi IP dei gateway sono 192.168.1.254 per la LAN A e 192.168.2.254 per la LAN B. I router IPsec sono distinti da ciascun gateway e usano due schede di rete: ad eth0 è assegnato un indirizzo IP statico accessibile esternamente, connesso ad Internet, mentre eth1 funge da punto di routing processando e trasmettendo i pacchetti della LAN da un nodo della rete ai suoi nodi remoti.

La connessione IPsec tra le due LAN, usa una chiave pre-condivisa di valore **r3dh4tl1nux**, e gli amministratori di A e B decidono di usare **racoon** per generare e condividere una chiave di autenticazione tra i router IPsec. L'amministratore della LAN A decide di chiamare la propria connessione **ipsec0**, mentre l'altro **ipsec1**.

Il seguente esempio, illustra il contenuto del file **ifcfg** per una connessione IPsec network-to-network sulla LAN A. Il nome univoco che identifica la connessione è ipsec0, cosicchè il file risultante è **/etc/sysconfig/network-scripts/ifcfg-ipsec0**.

```
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
SRCGW=192.168.1.254
DSTGW=192.168.2.254
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

I parametri contenuti nel file hanno il seguente significato:

TYPE=IPSEC
: Specifica il tipo di connessione

ONBOOT=yes
: Specifica se la connessione si avvia al boot del sistema

IKE_METHOD=PSK
: Specifica che la connessione usa il metodo di autenticazione pre-shared key (o chiave pre-condivisa).

SRCGW=192.168.1.254
: L'indirizzo IP del gateway locale. Per la LAN A, è il gateway della LAN A e per la LAN B, è il gateway della LAN B.

DSTGW=192.168.2.254
: L'indirizzo IP del gateway remoto. Per la LAN A, è il gateway della LAN B e per la LAN B è il gateway della LAN A.

SRCNET=192.168.1.0/24
: Specifica l'indirizzo della rete locale, che per questo esempio è l'indirizzo di rete della LAN A.

DSTNET=192.168.2.0/24
: Specifica l'indirizzo della rete remota, che per questo esempio è l'indirizzo di rete della LAN B.

DST=X.X.X.X
: L'indirizzo IP pubblico esternamente accessibile, sulla rete remota (LAN B).

L'esempio seguente riporta il contenuto del file della chiave pre-condivisa, **/etc/sysconfig/network-scripts/keys-ipsec*X*** (in cui *X* è 0 ed 1, rispettivamente, per le LAN A e B), usato da entrambe le reti per reciproca autenticazione. Il contenuto deve essere identico sulle due reti ed accessibile in lettura/scrittura soltanto all'utente root.

```
IKE_PSK=r3dh4tl1nux
```

Importante

Per modificare i permessi al file **keys-ipsec*X*** in modo che solo l'utente root possa leggere o modificare il file, usare il seguente comando:

```
chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

Per cambiare la chiave di autenticazione, modificare il file **keys-ipsec*X*** su entrambi i router di IPsec. *Le chiavi di autenticazione devono coincidere perchè la connessione funzioni correttamente*.

Di seguito si riporta il contenuto del file di configurazione **/etc/racoon/racoon.conf** per la connessione IPsec. Notare che il parametro **include** in basso, è inserito automaticamente ed è presente solo quando il tunnel IPsec è in esecuzione.

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
 pfs_group 2;
 lifetime time 1 hour ;
 encryption_algorithm 3des, blowfish 448, rijndael ;
 authentication_algorithm hmac_sha1, hmac_md5 ;
 compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf"
```

Ciò che segue sono le impostazioni specifiche per la connessione alla rete remota. Il file è denominato ***X.X.X.X*.conf** (dove *X.X.X.X* è l'indirizzo IP del router IPsec remoto). Notare che questo file è creato automaticamente all'attivazione del tunnel IPsec e non dovrebbe essere esplicitamente modificato.

```
remote X.X.X.X{
        exchange_mode aggressive, main;
 my_identifier address;
 proposal {
  encryption_algorithm 3des;
  hash_algorithm sha1;
  authentication_method pre_shared_key;
  dh_group 2 ;
 }
}
```

Prima di avviare la connessione IPsec, si dovrebbe abilitare l'IP forwarding nel kernel. Per abilitare l'IP forwarding, eseguire il seguente comando:

1. Modificare il file **/etc/sysctl.conf** impostando **net.ipv4.ip_forward** su **1**.

2. Usare il seguente comando per rendere effettiva la modifica:

```
[root@myServer ~] # sysctl -p /etc/sysctl.conf
```

Per avviare la connessione IPsec, usare il seguente comando su ciascun router:

```
[root@myServer ~] # /sbin/ifup ipsec0
```

A questo punto le connessioni sono attivate ed entrambe le LAN A e B possono comunicare tra loro. L'instradamento dei pacchetti è creato automaticamente dagli script di inizializzazione durante l'esecuzione di **ifup** sulla connessione IPsec. Per visualizzare un elenco di percorsi di instradamento, usare il comando:

```
[root@myServer ~] # /sbin/ip route list
```

Per testare la connessione IPsec, eseguire l'utility **tcpdump** sulla scheda di rete rivolta all'esterno (eth0 nel caso dell'esempio), che visualizza i pacchetti trasferiti tra gli host (o reti) e verifica se sono cifrati via IPsec. Per esempio, per verificare la connessione della LAN A, usare il comando:

```
[root@myServer ~] # tcpdump -n -i eth0 host lana.example.com
```

Il pacchetto dovrebbe includere un'intestazione AH ed essere segnato come ESP, ad indicare che si tratta di un pacchetto cifrato. Per esempio (le back slash denotano una continuazione di linea):

```
12:24:26.155529 lanb.example.com > lana.example.com: AH(spi=0x021c9834,seq=0x358): \
 lanb.example.com > lana.example.com: ESP(spi=0x00c887ad,seq=0x358) (DF) \
 (ipip-proto-4)
```

3.2.1.8. Avviare ed interrompere una connessione IPsec

Se la connessione IPsec, non è stata configurata per avviarsi al boot del sistema, si può usare un terminale da cui controllare l'avvio o l'interruzione.

Per avviare la connessione IPsec, usare il seguente comando su ciascun host per una connessione host-to-host, o router per una connessione network-to-network:

```
[root@myServer ~] # /sbin/ifup <nickname>
```

dove <*nickname*> è il nome precedentemente configurato, come **ipsec0**.

Per interrompere la connessione, usare il seguente comando:

```
[root@myServer ~] # /sbin/ifdown <nickname>
```

3.2.2. Secure Shell

SSH (Secure Shell), è un potente protocollo di rete usato per comunicare con altri sistemi attraverso un canale sicuro. Le trasmissioni su SSH sono cifrate e protette da intercettazioni. Può essere usato anche per accessi cifrati offrendo un metodo di autenticazione più robusto, rispetto ai tradizionali metodi basati su nome-utente e password.

SSH è molto semplice da attivare. Una volta avviato, il servizio sshd inizia ad accettare connessioni ed a permettere l'accesso al sistema solo dopo l'inserimento di un nome utente e password, corretti. Il numero di porta TCP standard del servizio SSH è 22; comunque può essere modificato nel file di configurazione **/etc/ssh/sshd_config**. Questo file contiene anche altre opzioni di configurazione di SSH.

Secure Shell (SSH) fornisce anche tunnel cifrati tra computer ma soltanto su una porta. *Il port forwarding può essere fatto usando un tunnel SSH*[1] ed il traffico può venir cifrato lungo il suo passaggio nel tunnel, tuttavia il port forwarding non è così fluido come con VPN.

3.2.3. Cifratura disco con LUKS

Lo standard Linux Unified Key Setup (o LUKS) cifra partizioni di disco di un sistema Linux. Ciò può risultare particolarmente importante nel caso dei portatili e dei supporti rimovibili. Inolltre LUKS consente l'uso di più chiavi utente per la decifrazione di una chiave principale, usata per cifrare la partizione.

3.2.3.1. Implementazione di LUKS in Fedora

Fedora 9 e le successive versioni, utilizzano LUKS per cifrare il file system. Per impostazione, l'opzione per cifrare il file system è disabilitata durante l'installazione di Fedora. Se il sistema viene installato con l'opzione di cifratura abilitata, allora ad ogni avvio del sistema verrà richiesto di inserire la frase di accesso (passphrase) per "sbloccare" la chiave di cifratura del disco. Se si decide di modificare la tabella di partizionamento predefinita, nelle impostazioni della tabella è possibile scegliere quali partizioni cifrare.

In Fedora, l'implementazione predefinita di LUKS si basa su AES 128 con funzione di hash SHA256. Gli algoritmi di cifratura disponibili sono:

- AES - Advanced Encryption Standard - *AES - FIPS PUB 197*[2]
- Twofish (Con blocco di cifratura da 128-bit)
- Serpent
- CAST-128 Encryption Algorithm - *RFC 2144*[3]
- CAST-256 Encryption Algorithm - *RFC 2612*[4]

3.2.3.2. Cifrare manualmente una Directory

Attenzione

Questa procedura comporta la rimozione completa dei dati dalla partizione da cifrare: tutti i dati contenuti nella partizione andranno PERSIi! Prima di procedere, assicurarsi di salvare i dati contenenti informazioni importanti su un supporto esterno!

Di seguito, si spiega come cifrare una partizione in una versione di Fedora corrente (e in versioni precedenti fino a Fedora 9); in particolare come cifrare la partizione **`/home`** (con altre partizioni il procedimento rimane lo stesso).

[1] http://www.redhatmagazine.com/2007/11/27/advanced-ssh-configuration-and-tunneling-we-dont-need-no-stinking-vpn-software
[2] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[3] http://www.ietf.org/rfc/rfc2144.txt
[4] http://www.ietf.org/rfc/rfc2612.txt

La seguente procedura cancella tutti i dati esistenti nella partizione: assicurarsi quindi, prima di iniziare, di aver adeguatamente salvato i propri dati importanti. Si richiede anche che sia presente una partizione separata per **`/home`** (p.e. /dev/VG00/LV_home). Inoltre tutti i comandi devono essere eseguiti come utente root. Se un qualche passaggio fallisce, non continuare ma risolvere il problema e riprendere la procedura soltanto a soluzione avvenuta.

3.2.3.3. Istruzioni passo passo

1. Accedere al runlevel 1: **`telinit 1`**
2. Smontare la partizione **`/home`** esistente: **`umount /home`**
3. In caso di falllimento usare il comando **`fuser`**, per trovare e terminare i processi che usano **`/home`**: **`fuser -mvk /home`**
4. Verificare che la partizione /home sia stata smontata: **`cat /proc/mounts | grep home`**
5. Scrivere la partizione con dati casuali: **`dd if=/dev/urandom of=/dev/VG00/LV_home`**. Il processo di scrittura può richiedere diverse ore.

   Importante

   Questo passaggio di riscrittura della partizione è un imperativo: assicura una buon protezione contro tentativi di intrusione. Dato che l'operazione richiede un lungo periodo per il suo completamento, si consiglia di effettuare questa operazione nei lunghi periodi di inutilizzo del sistema, per esempio durante la notte.

6. Inizializzare la partizione: **`cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`**
7. Aprire la partizione appena cifrata: **`cryptsetup luksOpen /dev/VG00/LV_home home`**
8. Verificare che esista la partizione: **`ls -l /dev/mapper | grep home`**
9. Creare un file system: **`mkfs.ext3 /dev/mapper/home`**
10. Montare la partizione: **`mount /dev/mapper/home /home`**
11. Verificare che la partizione sia visibile: **`df -h | grep home`**
12. Aggiungere al file /etc/crypttab la seguente riga: **`home /dev/VG00/LV_home none`**
13. Modificare il file /etc/fstab eliminando la riga relativa a /home ed aggiungendo la riga **`/dev/mapper/home /home ext3 defaults 1 2`**
14. Controllare la correttezza della riga inserita in fstab digitando: **`mount /home`**
15. Ripristinare i contesti di SELinux predefiniti: **`/sbin/restorecon -v -R /home`**
16. Riavviare il sistema: **`shutdown -r now`**
17. La riga precedentemente inserita in **`/etc/crypttab`** (al passo 12), richiede di inserire al boot la passphrase di **`luks`**.

18. Accedere come root e ripristinare il backup.

3.2.3.4. Risultato finale

Congratulazioni, ora si ha una partizione completamente cifrata che protegge con sicurezza tutti i dati a riposo, ossia a sistema spento.

3.2.3.5. Link di interesse

Per ulteriori informazioni su LUKS o sulla cifratura di dischi rigidi in Fedora, fare riferimento ai seguenti link:

- *LUKS - Linux Unified Key Setup*[5]
- *HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive, pvmove, and a Fedora LiveCD*[6]

3.2.4. Archivi 7-Zip cifrati

7-Zip[7] è uno strumento di compressione file, cross-platform di prossima generazione, usato per proteggere il contenuto degli archivi con un robusto sistema di cifratura (AES-256). Ciò è particolarmente utile per trasferire dati tra computer con sistemi operativi diversi (p.e. Linux a casa, windows in ufficio), essendo una soluzione di archiviazione con sistema di cifratura portabile.

3.2.4.1. Installazione di 7-Zip in Fedora

7-Zip non è un pacchetto base di Fedora ma può essere scaricato dal repository. Una volta installato il pacchetto riceverà gli aggiornamenti come avviene con gli altri pacchetti del sistema, senza richiedere particolare manutenzione.

3.2.4.2. Instruzioni di installazione passo passo

- Aprire un terminale: p.e. in GNOME, selezionare **`''Applicazioni'' -> ''Strumenti di Sistema'' -> ''Terminale''`**
- Installare 7-Zip come utente root: **`sudo yum install p7zip`**
- Chiudere il terminale: **`Ctrl+D`**

3.2.4.3. Istruzioni d'uso passo passo

Di seguito si riportano le istruzioni per comprimere e cifrare la propria cartella **`Documenti`**. La cartella **`Documenti`** originaria, rimane inalterata. Questa tecnica si può applicare a tutte le altre cartelle o file del sistema a cui si ha accesso. Si presume di lavorare in ambiente GNOME.

- Aprire un terminale: selezionare **`''Applicazioni'' -> ''Strumenti di Sistema'' -> ''Terminale''`**
- Comprimere e Cifrare: (inserire una password quando richiesto) **`7za a -mhe=on -ms=on -p Documenti.7z Documenti/`**

[5] http://code.google.com/p/cryptsetup/
[6] https://bugzilla.redhat.com/attachment.cgi?id=161912
[7] http://www.7-zip.org/

La cartella **`Documenti`** è ora compressa e cifrata. Successivamente si sposta la cartella archivio da un'altra parte, dove verrà estratta.

- Creare una nuova directory: **`mkdir nuovaDirectory`**
- Spostare la cartella archivio nella nuovaDirectory: **`mv Documenti.7z nuovaDirectory`**
- Spostarsi nella nuovaDirectory: **`cd nuovaDirectory`**
- Estrarre i file: (inserire la password, quando richiesto) **`7za x Documenti.7z`**

I file estratti dall'archivio ora si trovano nella nuovaDirectory. Le seguenti istruzioni ripristinano le condizioni iniziali, rimuovendo tutti i file e directory create.

- Spostarsi nella directory superiore: **`cd ..`**
- Eliminare la cartella nuovaDirectory, contenente l'archivio e i file estratti: **`rm -rf nuovaDirectory`**
- Chiudere il terminale: **`Ctrl+D`**

3.2.4.4. 7-Zip e gli altri sistemi operativi

7-Zip, per impostazione, non viene distribuito con microsoft windows o mac os x. Se si vuole usare 7-Zip su queste piattaforme occorre *scaricare*[8] le versioni appropriate a questi sistemi operativi.

3.2.5. Usare GNU Privacy Guard (GnuPG)

GnuPG (GPG) è usato per identificare gli utenti ed autenticare le comunicazioni, incluse quelle con persone non direttamente note. GPG consente a chi riceve una email firmata GPG di verificare l'autenticità del messaggio. In altre parole, GPG garantisce con ragionevole certezza che le comunicazioni firmate provengono effettivamente da chi ha le ha firmate. GPG è utile perchè impedisce a un terzo (l'intruso) di alterare il messaggio, intercettare conversazioni o corrompere codice.

GPG può essere usato anche per firmare e/o cifrare i file sul proprio sistema o su un drive di rete. Ciò serve ad aumentare la protezione impedendo che un file venga alterato o letto da persone non autorizzate.

Per poter usare GPG per autenticare o cifrare email occorre dapprima creare una coppia di chiavi, pubblica e privata. Una volta create, per poterle utilizzare occorre impostare il client di posta.

3.2.5.1. Generare chiavi GPG in GNOME

Installare l'utility **Seahorse**, un'applicazione che semplifica la gestione delle chiavi GPG. Dal menu, selezionare **Sistema > Amministrazione > Aggiungi/Rimuovi Software**, per avviare PackageKit. Nella casella di testo, inserire *Seahorse* e poi premere Trova. Abilitare la casella accanto al pacchetto "seahorse" e poi premere **Applica** per avviare il processo di installazione (verifica dipendenze, scaricamento, verifica integrità, installazione). *Seahorse* può essere installato anche da terminale, digitando **`su -c "yum install seahorse"`**.

Per creare una chiave, avviare **`Seahorse`** selezionando **Applicazioni > Accessori > Password e chiavi di cifratura**. Dal menu della finestra principale, selezionare **File > Nuovo..." > Chiave PGP** e poi premere **Continua**. Inserire il nome e il cognome, l'indirizzo email ed opzionalmente un commento

[8] http://www.7-zip.org/download.html

(es. pippo, pippo@paperopoli.org, L'amico di Topolino e il fidanzato di Clarabella). Poi premere **Crea**. Nella nuova finestra di dialogo inserire una frase d'accesso (passphrase) per la chiave. Scegliere una passphrase, robusta e facile da ricordare. Infine premere **Ok** per creare la chiave.

Attenzione

Se si dimentica la passphrase la chiave non può più essere usata e tutti i dati cifrati con la chiave andranno perduti.

L'ID della chiave GPG si trova nella colonna **ID chiave** della finestra principale di Seahorse, accanto al **Nome** della chiave creata. In molti casi quando viene richiesta la ID chiave, occorre anteporre al valore di ID, il prefisso "0x", come in "0x6789ABCD". Si raccomanda di creare una copia di backup della chiave e di custodirla in un luogo sicuro e protetto.

3.2.5.2. Generare chiavi GPG in KDE

Avviare il programma KGpg, selezionando **Applications > Utilities > Encryption Tool**. Se è la prima volta che si usa KGpg, il programma avvia un wizard da cui creare una coppia di chiavi GPG. Occorre inserire il nome, l'indirizzo di posta ed un commento (opzionale). Si può indicare anche una scadenza per la chiave, come pure il grado di robustezza (numero di bit) e l'algoritmo di cifratura. Nella seconda pagina del wizard si richiede di inserire una passphrase, per poter usare la chiave. Al termine del processo di crezione la chiave compare nella finestra principale di **`KGpg`**.

Attenzione

Se si dimentica la passphrase la chiave non può più essere usata e tutti i dati cifrati con la chiave andranno perduti.

L'ID della chiave GPG si trova nella colonna **ID chiave** della finestra principale di Seahorse, accanto al **Nome** della chiave creata. In molti casi quando viene richiesta la ID chiave, occorre anteporre al valore di ID, il prefisso "0x", come in "0x6789ABCD". Si raccomanda di creare una copia di backup della chiave e di custodirla in un luogo sicuro e protetto.

3.2.5.3. Generare chiavi GPG con un terminale

Usare il seguente comando di shell: **`gpg --gen-key`**

Il comando genera una coppia di chiavi, una pubblica ed una privata. I destinatari usano la chiave pubblica per autenticare e/o decifrare le comunicazioni. Distribuire la chiave pubblica alle persone interessate a ricevere comunicazioni autenticate come le mailing list. Il Fedora Documentation Project, per esempio, richiede ai propri partecipanti di indicare la propria chiave GPG nelle propria pagina personale.

Una serie di prompt condurrano lungo processo di creazione. Per assegnare valori predefiniti basta premere il tasto **`Invio`**. Il primo prompt richiede di selezionare il tipo di chiave:

Selezionare un tipo di chiave: (1) DSA and ElGamal (predefinto) (2) DSA (solo firma) (4) RSA (solo firma). Nella maggior parte dei casi il valore predefinito va bene. Infatti una chiave DSA/ElGamal consente non solo di firmare le comunicazioni ma anche di cifrare file.

Poi inserire la lunghezza della chiave: la lunghezza minima è 768 bit; 1024 bit la lunghezza predefinita e 2048 la massima lunghezza. Di nuovo il valore predefinito è sufficiente per la maggior parte degli utenti e rappresenta un livello di sicurezza *estremamente* robusto.

Poi selezionare la durata della chiave. E' una buona idea impostare una data di scadenza invece di usare il valore predefinito che è "none". Se per esempio, l'indirizzo email coperto dalla chiave non è più valido, una data di scadenza avviserà i destinatari di non usare più quella chiave pubblica.

Specificare la durata della chiave; 0: nessuna scadenza; d: scadenza dopo d giorni; w: scadenza dopo w settimane; m: scadenza dopo m mesi; y: scadenza dopo y anni.

Inserendo per esempio **`1y`**, la chiave avrà validità di un anno. (Tenere presente che è possibile modificare la scadenza anche successivamente).

Prima di richiedere altre informazioni, appare il seguente prompt: **`Is this correct (y/n)?`** Inserire **`y`**, per terminare il processo.

Successivamente, inserire il proprio nome ed indirizzo email. Ricordare che il processo di creazione di una chiave pubblica serve ad identificare se stessi come persone reali, inserire perciò il proprio nome reale. Non usare alias o nickname che potrebbero mascherare la propria identità.

Inserire il proprio indirizzo email reale. Se si inserisce un indirizzo fasullo gli altri potrebbero avere dei problemi a rintracciare la chiave pubblica e potrebbe complicare l'autenticazione delle comunicazioni. Se per esempio la chiave GPG è impiegata per far parte della mailing list del Docs Project, inserire la email usata per accedere alla mailing list.

Nel campo commento inserire alias o altre informazioni a piacere. (Alcune persone usano chiavi differenti per scopi differenti, identificando ciascuna chiave con un commento, come "Ufficio" o "Fedora Project").

Al prompt di conferma, se tutte le informazioni sono corrette, digitare O per continuare o usare le altre opzioni per risolvere eventuali problemi. Infine inserire una passphrase per proteggere la propria chiave segreta. Il programma **gpg** richiede di inserire due volte in successione la stessa passphrase, scongiurando errori di battitura.

A questo punto, **gpg** genera dei dati random garantendo una chiave segreta (pressocchè) unica. Per aiutare l'applicazione a migliorare la generazione random dei dati può essere efficace durante questa fase, spostare il mouse, digitare sulla tastiera o fare altre operazioni. Una volta completato questo passaggio, le chiavi sono pronte per l'uso:

```
pub  1024D/1B2AFA1C 2005-03-31 luigi votta (Fedora Docs Project) <lewis41@fedoraproject.org>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

Key fingerprint (impronta digitale) è una breve "firma" della propria chiave. Essa permette di confermare ai destinatari di aver ricevuto la chiave senza alcuna manomissione. Non occorre ricordare la propria fingerprint. Per visualizzarla basta usare il comando **`gpg --fingerprint lewis41@fedoraproject.org`**.

La "GPG key ID" (ID della chiave GPG) è composta da 8 numeri esadecimali (base 16: 0-F). Nell'esempio precedente l'ID della chiave GPG, è pari a 1B2AFA1C. In molte situazioni quando viene richiesto il proprio ID della chiave GPG, occorre anteporre il simbolo "0x" all'ID, come in "0x1B2AFA1C".

Attenzione

Se si dimentica la passphrase la chiave non può più essere usata e tutti i dati cifrati con la chiave andranno perduti.

3.2.5.4. Usare GPG con Alpine

Se si usa il client di posta **Alpine** o **Pine**, per usare GPG occorre installare il pacchetto *ez-pine-gpg* scaricabile da *http://business-php.com/opensource/ez-pine-gpg/*. Una volta installato, occorre modificare il file **`~/.pinerc`**. Ossia:

1. il path **`/home/username/bin`** deve essere sostituito con il path del pacchetto installato

2. individuare i due gpg-identifier dopo la stringa _RECIPIENTS_, e sostituirli con l'ID della chiave GPG. In questo modo spedendo un messaggio cifrato a qualcuno, il messaggio viene cifrato anche con la propria chiave; senza questa impostazione non sarebbe possibile leggere il messaggio nella cartella dei messaggi inviati.

La modifica dovrebbe assomigliare a qualcosa di simile:

```
# This variable takes a list of programs that message text is piped into
# after MIME decoding, prior to display.
display-filters=_LEADING("-----BEGIN PGP")_ /home/max/bin/ez-pine-gpg-incoming

# This defines a program that message text is piped into before MIME
# encoding, prior to sending
sending-filters=/home/max/bin/ez-pine-gpg-sign _INCLUDEALLHDRS_,
    /home/username/bin/ez-pine-gpg-encrypt _RECIPIENTS_ gpg-identifier,
    /home/username/bin/ez-pine-gpg-sign-and-encrypt _INCLUDEALLHDRS_ _RECIPIENTS_ gpg-
identifier
```

3.2.5.5. Usare GPG con Evolution

3.2.5.5.1. Configurare GPG per l'uso con Evolution

Per configurare GPG in **Evolution**, dal menu di **Evolution** selezionare **Modifica > □□□ > Preferenze**. Nella finestra delle **Preferenze di Evolution**, selezionare nel pannello di sinistra **Account di posta**. Nel pannello di destra selezionare l'account di posta che si vuole autenticare. Poi premere il pulsante **Modifica**. Nella finestra delle impostazioni **Editor account**, selezionare la scheda **Sicurezza**.

Nel campo di testo etichettato **ID della chiave PGP/GPG**, inserire l'ID della chiave GPG corrispondente a questo account di posta. Un metodo per scoprire l'ID della chiave è usare questo comando in un terminale: **`gpg --fingerprint EMAIL_ADDRESS`**. L'ID della chiave coincide con gli ultimi otto caratteri (4 byte) del fingerprint della chiave. Può essere una buona idea abilitare anche la casella con l'etichetta **Cifrare sempre per se stessi quando si inviano messaggi cifrati**. Si potrebbe anche abilitare la casella **Firmare sempre i messaggi in uscita quando si usa questo account**.

Nota

Se le chiavi pubbliche non vengono contrassegnate come fidate non sarà possibie cifrare le email, a meno di non selezionare l'opzione **Dare sempre fiducia nel cifrare alle chiavi nel portachiavi personale**. In tal caso si riceve un messaggio in cui si segnala il fallimento della verifica di fiducia.

3.2.5.5.2. Verificare le email con Evolution

Evolution verifica automaticamente la validità di ogni messaggio ricevuto. Se Evolution non riesce a verificare la firma GPG di un messaggio a causa di una chiave pubbilca mancante (o manomessa), nella parte in basso del messaggio compare una banda rossa. Se il messaggio è stato verificato ma la chiave non risulta firmata nè localmente nè globalmente, il banner è di colore giallo. Se il messaggio è stato verificato e la chiave risulta firmata, il banner è verde. Cliccando sull'icona con il sigillo all'interno del banner, Evolution visualizza una finestra con informazioni di sicurezza sulla firma. Per aggiungere una chiave pubblica al proprio porta chiavi personale, usare la funzione di ricerca e l'indirizzo email del proprietario della chiave: **`gpg --keyserver pgp.mit.edu --search email address`**. Per importare la chiave corretta occorre che l'ID della chiave coincida con le informazioni fornite da Evolution.

3.2.5.5.3. Firmare e cifrare email con Evolution

Firmare una email consente al destinatario di verficare l'autenticità della email, ossia del mittente. Il Fedora Project incoraggia caldamente i propri utenti a firmare le email, incluse quelle indirizzate alle mailing list dei vari progetti Fedora. Cifrare le email consente di leggere il loro contenuto soltanto ai destinatari, per questo motivo non cifrare le email inviate alle mailing list.

Nelle impostazioni dell'account selezionare la scheda **Sicurezza**. Per firmare le proprie email inserire nella casella di testo con l'etichetta **ID della chiave PGP/GPG**, l'ID della propria chiave. Per cifrare le email, abilitare la casella con l'etichetta **Cifrare sempre per stessi quando si inviano messaggi cifrati**. Un messaggio cifrato può anche essere firmato ed è una buona regola farlo. Al momento dell'invio di una email firmata Evolution richiede di inserire la passphrase per la chiave GPG (dopo tre tentativi falliti Evolution segnala un messaggio di errore). Se si abilita la casella con l'etichetta **Ricorda la password per il resto della sessione**, non occorrere reinserire la passphrase per firmare o decifrare email nelle volte successive, a meno di non chiudere e riavviare una nuova sessione.

3.2.5.6. Usare GPG con Thunderbird

Fedora include Mozilla Thunderbird nel pacchetto *thunderbird*, ed il pacchetto *mozilla-mail* contenente l'applicazione di posta di Mozilla. Thunderbird è il client di posta raccomandato di Mozilla. Thunderbird è accessibile da **Applicazioni > Internet > Thunderbird Email**.

I prodotti Mozilla supportano varie estensioni, componenti che aggiungono nuove funzionalità alle applicazioni principali. Le estensioni Enigmail offrono supporto GPG ai client di posta di Mozilla. Esistono versioni di Enigmail sia per Mozilla Thunderbird sia per Mozilla Suite (Seamonkey). Il software Netscape di AOL è basato sui prodotti Mozilla e può usare queste estensioni.

Per installare Enigmail su Fedora seguire le seguenti istruzioni.

Enigmail usa il termine OpenPGP nei menu e tra le opzioni. GPG è una implementazione di OpenPGP ed entrambe le terminologie possono considerarsi equivalenti.

Enigmail si può scaricare da *http://enigmail.mozdev.org/download.html*.

Per screenshot sull'impiego di Enigmail e GPG visitare *http://enigmail.mozdev.org/screenshots.html*.

3.2.5.6.1. Installazione di Enigmail

Enigmail è anche disponibile nei repository di Fedora e può essere installato usando il comando **`yum install thunderbird-enigmail`** in un terminale. In alternativa si può procedere con l'ausilio grafico del Gestore dei pacchetti, selezionando **Sistema -> Amministrazione -> Aggiungi/Rimuovi Software** dal menu principale, e installando il pacchetto denominato *thunderbird-enigmail*.

3.2.5.7. Sulla crittografia a chive pubblica

1. *Wikipedia - Crittografia asimmetrica*[9]

2. *How Encryption Works*[10]

[9] http://it.wikipedia.org/wiki/Crittografia_asimmetrica
[10] http://computer.howstuffworks.com/encryption.htm

Principi generali di Sicurezza dell'Informazione

I seguenti principi generali offrono una panoramica sulle buone pratiche di sicurezza:

- cifrare i dati trasmessi in rete per ridurre gli attacchi tipo man-in-the-middle e le possibilità di intercettazione. E' particolarmente importante cifrare le informazioni di autenticazione come le password.
- minimizzare la quantità di software installato e dei servizi in esecuzione.
- usare software e strumenti che aumentino la sicurezza come Security-Enhanced Linux (SELinux) per controlli MAC (Mandatory Access Control), iptables di Netfilter per il filtraggio di pacchetti (firewall) e GNU Privacy Guard (GnuPG) per cifrare file.
- eseguire se possibile, ogni servizio di rete su un server differente per minimizzare il rischio che la compromissione di un servizio possa essere usato per compromettere anche altri servizi.
- mantenere gli account utenti: creare e rinforzare la policy delle password; eliminare gli account utente non usati.
- controllare regolarmente i log di sistema e delle applicazioni. Per impostazione, gli avvisi (log) di sistema relativi alla sicurezza sono salvati nei file **`/var/log/secure`** e **`/var/log/audit/audit.log`**. Nota: la trasmissione dei log su un server dedicato serve ad impedire che gli attaccanti possano facilmente modificare i log locali eliminando le tracce dei loro tentativi di intrusione.
- non accedere mai direttamente come root a meno che non sia assolutamente necessario. Gli amministratori dovrebbero usare **`sudo`** per eseguire comandi root. Gli account che possono usare **`sudo`** sono specificati in **`/etc/sudoers`**. Usare lo strumento **`visudo`** per modificare il file **`/etc/sudoers`**.

4.1. Consigli, guide e strumenti

L'agenzia statunitense *NSA*[1] (National Security Agency), fornisce fondamentali guide e consigli per molti sitemi operativi, per aiutare le agenzie governative, le aziende e gli individui a rendere sicuri i propri sistemi da attacchi informatici. Per esempio, le seguenti guide in formato PFD, sono dedicate al sistema Red Hat Enterprise Linux 5:

- *Hardening Tips for the Red Hat Enterprise Linux 5*[2]
- *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*[3]

L'agenzia *DISA*[4] (Defense Information Systems Agency), fornisce documenti, checklist e test (*I.A.S.E.*[5] o Information Assurance Support Environment), che aiutano a rendere sicuro il proprio sistema. *U.S.T.I.G.*[6] (pdf) o Unix Security Technical Implementation Guide, è una guida sulla sicurezza in UNIX - una guida per utenti avanzati di UNIX e Linux.

---

[1] www.nsa.gov
[2] http://www.nsa.gov/ia/_files/os/redhat/rhel5-pamphlet-i731.pdf
[3] http://www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf
[4] http://www.disa.mil/
[5] http://iase.disa.mil/index2.html
[6] http://iase.disa.mil/stigs/stig/unix-stig-v5r1.pdf

Il pacchetto *UNIX Security Checklist Version 5, Release 1.26*[7] fornito dalla DISA, contiene una raccolta di documenti e checklist che vanno dai permessi da assegnare ai file ai controlli da fare sul sistema.

Inoltre, la DISA ha reso disponibile degli script *UNIX SPR*[8] che permettono agli amministratori di controllare specifiche impostazioni di sistema. Questi script elencano in un rapporto, in formato XML, tutte le vulnerabilità note presenti nel sistema.

[7] http://iase.disa.mil/stigs/downloads/zip/unclassified_unix_checklist_v5r1-26_20100827.zip
[8] http://iase.disa.mil/stigs/SRR/unix.html

Installazione sicura

La sicurezza inizia nel momento in cui si inserisce il CD o DVD nel lettore per installare Fedora. Configurare il sistema in modo sicuro dall'inizio, semplifica l'implementazione di ulteriori impostazioni di sicurezza successive.

5.1. Partizioni del disco

L'NSA raccomanda di creare partizioni separate per /boot, /, /home, /tmp e /var/tmp. Le motivazioni di questa scelta sono le seguenti:

/boot - Questa partizione è la prima ad essere letta dal sistema durante la fase di avvio del sistema. Il boot loader e le immagini kernel usate per avviare il sistema Fedora, si trovano in questa partizione. La partizione non dovrebbe essere cifrata. Se i dati di questa partizione fossero inclusi in / e quest'ultima venisse cifrata o diventasse inutilizzabile allora il sistema non sarebbe capace di avviarsi.

/home - Se i dati utente si trovassero in / invece che in una partizione separata, la partizione si riempirebbe a tal punto da portare all'instabilità del sistema operativo. Inoltre, l'up-grade del sistema è molto più semplice se i dati utente si trovano nella proporia partizione di /home, in quanto essi non vengono modificati durante l'aggiornamento di Fedora. Inoltre, se la partizione / si corrompe tutti i dati utente potrebbero, molto probabilmente, andare perduti per sempre. Invece una partizione separata garantisce una migliore protezione contro la perdita dei dati. In tal modo si possono anche programmare backup regolari di questa partizione.

/tmp e /var/tmp - Sia la directory /tmp sia la directory /var/tmp sono usate per contenere i dati temporanei, cioè che non hanno una lunga durata. Inoltre, se un flusso di dati satura una di queste directory esso potrebbe riempire tutto lo spazio disponibile. In tal caso e se le directory si trovassero in / il sistema diventerebbe presto instabile e ci sarebbe un crash. Per questo motivo è una buona idea realizzare partizioni separate per queste directory.

5.2. Utilizzo di LUKS

A partire da Fedora 9 l'implementazione del sistema di cifratura del disco, *LUKS*[1] (Linux Unified Key Setup), è diventato più semplice. Durante il processo di installazione l'utente ha la possibilità di cifrare le proprie partizioni. L'utente deve fornire una passphrase che sarà la chiave per sbloccare la chiave di cifratura usata per rendere più sicuri i dati della partizione.

[1] http://fedoraproject.org/wiki/Security_Guide/9/LUKSDiskEncryption

Manutenzione del software

La manutenzione del software è estremente importante per mantenere sicuro un sistema. E' di vitale importanza applicare patch (correzioni) ai programmi appena si rendono disponibili, in modo da impedire agli attaccanti di sfruttare le falle scoperte per infiltrarsi nel sistema.

6.1. Installare il software indispensabile

E' una buona pratica installare soltanto i pacchetti dei programmi usati, dato che ogni pezzo di codice potrebbe contenere una vulnerabilità. Se si installa da un DVD si ha la possibilità di selezionare esattamente i pacchetti da installare. Poi una volta installato il sistema, se si ha la necessità di altri programmi essi possono sempre essere installati successivamente.

6.2. Pianificare e configurare gli aggiornamenti di sicurezza

Il software in generale contiene bug. Spesso, questi bug possono risultare in una vulnerabilità tale da esporre il sistema agli attacchi di utenti maliziosi. I sistemi non aggiornati con patch di sicurezza sono una causa comune di intrusione. Si dovrebbe pianificare di installare, con regolarità, patch di siucrezza per rimuovere tali vulnerabilità.

Per gli utenti domestici gli aggiornamenti di sicurezza dovrebbero essere installati appena possibile. Configurare l'installazione automatica degli aggiornamenti di sicurezza è un modo per evitare di dimenticarsene, anche se talvolta può comportare il rischio che si possano creare conflitti con la configurazione o altri software nel sistema.

Per gli utenti business o gli utenti domestici con esperienza, gli aggiornamenti di siucrezza dovrebbero essere testati e programmati. Ulteriori misure a protezione del sistema dovrebbero essere prese durante il periodo tra il rilascio delle patch e la loro installazione. Queste misure dipenderanno dal rischio effettivo della vulnerabilità e potrebbero includere regole di firewall aggiuntive, l'uso di firewall esterni o modifche alle impostazioni software.

6.3. Regolare gli aggiornamenti automatici

Fedora è configurato per applicare gli aggiornamenti su base giornaliera. Per modificare questa impostazione occorre aprire la finestra **Preferenze di aggiornamento**. E' possibile impostare ogni quanto tempo controllare la disponibilità di aggiornamenti, il tipo di aggiornamenti da applicare e se avvisare o meno della disponibilità di aggiornamenti.

In GNOME, i controlli per gli aggiornamenti si trovano selezionando **`Sistema -> Preferenze -> Aggiornamento Software`**. In KDE, si trovano selezionando: **`Applications -> Settings -> Software Updates`**.

6.4. Installare pacchetti firmati da repository fidati

I pacchetti software sono resi pubblici attraverso repository. Tutti i repository fidati supportano la firma dei pacchetti che usa la tecnologia a chiave pubblca per garantire che i pacchetti pubblicati nel repository non abbiano subito manomissioni dal momento della loro firma. Ciò serve a evitare di installare software che potrebbe essere stato maliziosamente alterato in seguito alla sua pubblicazione.

Usare troppi repository, repository non fidati o repository con pacchetti privi di firma aumenta il rischio di introdurre nel proprio sistema, codice malizioso o vulnerabile. Aggiungere con prudenza i repository al gestore del software **yum**.

Common Vulnerabilities and Exposures

Il sistema CVE o Common Vulnerabilities and Exposures (Vulnerabilità ed Esposizioni Comuni), offre un sistema di riferimento per vulnerabilità e falle di sicurezza note pubblicamente. ITRE Corporation gestisce il sistema con fondi del National Cyber Security Division del Department of Homeland Security degli Stati Uniti d'America.

MITRE Corporation assegna un identificatore CVE ad ogni vulnerabilità o falla di sicurezza. Il CVE è usato per tracciare la vulnerabilità nei vari pezzi di codice, dato che una singolo CVE può interessare diversi pacchetti software e diversi rivenditori.

7.1. Plugin YUM

Il pacchetto *yum-plugin-security* è una caratteristica di Fedora. Se installato, questo modulo di yum fa in modo di recuperare soltanto gli aggiornamenti di sicurezza. Può essere usato anche per fornire informazioni sull'avviso Red Hat, sul bug nel database di Bugzilla Red Hat o sul numero di CVE dalla directory del MITR, cui fa riferimento l'aggiornamento di un pacchetto.

Per abilitare questa caratterisitica basta semplicemente installare il plugin con il comando **`yum install yum-plugin-security`**.

7.2. Usare yum-plugin-security

Il principale comando di questo plugin è **`yum list-sec`**. E' molto simile a **`yum check-update`** con la differenza che elenca anche l'ID di Red Hat dell'avviso ed il tipo di ciascun aggiornamento come “enhancement” (miglioramento), “bugfix” (risoluzione) o “security” (sicurezza):

RHSA-2007:1128-6 security autofs - 1:5.0.1-0.rc2.55.el5.1.i386
RHSA-2007:1078-3 security cairo - 1.2.4-3.el5_1.i386
RHSA-2007:1021-3 security cups - 1:1.2.4-11.14.el5_1.3.i386
RHSA-2007:1021-3 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386

Se si usa **`yum list-sec cves`**, l'ID Red Hat è rimpiazzato dall'ID in CVE dell'avviso cui fa riferimento l'aggiornamento; se si usa **`yum list-sec bzs`** l'ID si riferisce a quello in Bugzilla di Red Hat. Se un pacchetto si riferisce a ID multipli in Bugzilla o CVE, il pacchetto potrebbe essere elencato più volte:

Ecco un tipico esempio d'output di **`yum list-sec bzs`**:
410031 security autofs - 1:5.0.1-0.rc2.55.el5.1.i386
387431 security cairo - 1.2.4-3.el5_1.i386
345101 security cups - 1:1.2.4-11.14.el5_1.3.i386
345111 security cups - 1:1.2.4-11.14.el5_1.3.i386
345121 security cups - 1:1.2.4-11.14.el5_1.3.i386
345101 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386
345111 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386
345121 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386

Un esempio d'output di **`yum list-sec cves`**:
CVE-2007-5964 security autofs - 1:5.0.1-0.rc2.55.el5.1.i386
CVE-2007-5503 security cairo - 1.2.4-3.el5_1.i386
CVE-2007-5393 security cups - 1:1.2.4-11.14.el5_1.3.i386
CVE-2007-5392 security cups - 1:1.2.4-11.14.el5_1.3.i386
CVE-2007-4352 security cups - 1:1.2.4-11.14.el5_1.3.i386
CVE-2007-5393 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386

```
CVE-2007-5392 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386
CVE-2007-4352 security cups-libs - 1:1.2.4-11.14.el5_1.3.i386
```

L'altro comando disponibile in *yum-plugin-security* è **`info-sec`**. Esso accetta un numero d'avviso come argomento, un ID CVE o Bugzilla e restituisce informazioni dettagliate sull'avviso, inclusa una breve argomentazione sulla natura del problema o dei problemi sollevati dall'avviso.

Oltre a questi due nuovi comandi sono disponibili anche nuove opzioni nel comando **`yum update`**, per selezionare solo aggiornamenti di sicurezza o solo aggiornamenti associati ad un avviso o bug.

Per applicare solo aggiornamenti di sicurezza, usare:
`yum update --security`

Per applicare tutti gli aggiornamenti al Bug #410101 di Bugzilla, eseguire:
`yum update --bz 410101`

Per applicare tutti gli aggiornamenti relativi all'avviso di CVE con ID CVE-2007-5707 e gli aggiornamenti relativi all'avviso di Red Hat con ID RHSA-2007:1082-5, eseguire:
`yum update --cve CVE-2007-5707 --advisory RHSA-2007:1082-5`

Maggiori informazioni su queste nuove funzioni sono documentate nelle pagine di man su *yum-plugin-security*(8).

Per maggiori informazioni sugli aggiornamenti di sicurezza in Fedora, si prega di visitare la pagina Fedora Security al seguente link *https://fedoraproject.org/wiki/Security*.

Riferimenti

I seguenti riferimenti sono collegamenti ad ulteriori informazioni rilevanti in SELinux e Fedora ma che esulano dagli scopi di questa guida. Notare che dato il rapido sviluppo di SELinux, alcuni materiali potrebbero applicarsi solo a specifiche versioni di Fedora.

Libri

SELinux by Example

Mayer, MacMillan, and Caplan

Prentice Hall, 2007

Tutorial ed aiuto

Understanding and Customizing the Apache HTTP SELinux Policy

http://docs.fedoraproject.org/selinux-apache-fc3/

Tutorials and talks from Russell Coker

http://www.coker.com.au/selinux/talks/ibmtu-2004/

Generic Writing SELinux policy HOWTO

http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

Red Hat Knowledgebase

http://kbase.redhat.com/

Informazioni generali

Sito web NSA SELinux

http://www.nsa.gov/selinux/[1]

NSA SELinux FAQ

http://www.nsa.gov/selinux/info/faq.cfm[2]

Fedora SELinux FAQ

http://docs.fedoraproject.org/en-US/Fedora/13/html/SELinux_FAQ/index.html

SELinux NSA's Open Source Security Enhanced Linux

http://www.oreilly.com/catalog/selinux/

Tecnologia

Una introduzione a Object Classes and Permissions

http://www.tresys.com/selinux/obj_perms_help.html

Integrating Flexible Support for Security Policies into the Linux Operating System (una retrospettiva sull'implementazione di Flask in Linux)

http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

Implementing SELinux as a Linux Security Module

http://www.nsa.gov/research/_files/publications/implementing_selinux.pdf

A Security Policy Configuration for the Security-Enhanced Linux

http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

[1] http://www.nsa.gov/research/selinux/index.shtml
[2] http://www.nsa.gov/research/selinux/faqs.shtml

Community

Fedora SELinux User Guide
: *http://docs.fedoraproject.org/selinux-user-guide/*

Fedora SELinux Managing Confined Services Guide
: *http://docs.fedoraproject.org/selinux-managing-confined-services-guide/*

SELinux community page
: *http://selinux.sourceforge.net*

IRC
: irc.freenode.net, #selinux, #fedora-selinux, #security

Storia

Quick history of Flask
: *http://www.cs.utah.edu/flux/fluke/html/flask.html*

Full background on Fluke (Flux µ-kernel Environment)
: *http://www.cs.utah.edu/flux/fluke/html/index.html*

Appendice A. Standard di crittografia

A.1. Crittografia sincrona

A.1.1. Advanced Encryption Standard - AES

In crittografia, lo standard AES (Advanced Encryption Standard) è un algoritmo di cifratura standard adottato dal governo degli Stati Uniti d'America. Lo standard prevede tre blocchi di cifratura, AES-128, AES-192 e AES-256, adottati da una collezione più larga originariamente nota come Rijndael. Ciascuna blocco di cifratura di 128 bit ha chiavi da 128, 192 e 256 bit, rispettivamente. Le cifrature AES sono state ampiamente analizzate e ora sono usate in tutto il mondo in sostituzione del suo predecessore il DES (Data Encryption Standard).[1]

A.1.1.1. Usi dell'AES

A.1.1.2. Storia dell'AES

L'AES è stato annunciato dal NIST (National Institute of Standards and Technology), nel U.S. FIPS PUB 197 (FIPS 197) il 26 novembre del 2001, dopo un periodo di standardizzazione durato cinque anni, in cui quindici progetti alternativi sono stati analizzati e studiati, riconoscendo il Rijndael come il più adatto (vedere il processo di sviluppo dell'Advanced Encryption Standard, per maggiori dettagli). L'AES è divenuto uno standard effettivo il 26 maggio 2002. E' disponibile in diversi pacchetti di cifratura. L'AES è il primo algoritmo di cifratura pubblicamente accessibile ed aperto, approvato dall'NSA per proteggere informazioni top secret. [2]

L'algoritmo di cifratura Rijndael è stato progettato da due progettisti belgi, Joan Daemen e Vincent Rijmen. Il nome Rijndael è una parola composta da parti di nome dei due inventori.[3]

A.1.2. Data Encryption Standard - DES

Lo standard DES (Data Encryption Standard), è un cifrario a blocchi, scelto dal National Bureau of Standards degli Stati Uniti d'America, come standard per cifrare le informazioni delle agenzie federali (o FIPS: Federal Information Processing Standard), a partire dal 1976 e poi adottato globalmente da altri Stati. Il DES si basa su un algoritmo di cifratura a chiave simmetrica di 56 bit. L'algoritmo fin dai suoi esordi presentava diverse difficoltà nei suoi elementi progettuali con una chiave relativamente corta e il sospetto di manomissioni da parte dell'NSA (National Security Agency). Conseguentemente il DES divenne oggetto di approfondite anailsi da parte di numerose università che portarono alle attuali conoscenze sugli algoritmi di crittografia e sulle tecniche di crittoanalisi.[4]

A.1.2.1. Usi del DES

A.1.2.2. Storia del DES

Il DES è ufficialmente riconosciuto come insicuro per molte applicazioni, principlamente a causa della scarsa lunghezza della chiave, 56 bit. Nel gennaio 1999 due agenzie, la Distributed.net e la Electronic

[1] "Advanced Encryption Standard" *Wikipedia.* 28 sett 2010 *http://it.wikipedia.org/wiki/Advanced_Encryption_Standard*
[2] "Advanced Encryption Standard" *Wikipedia.* 28 sett 2010 *http://it.wikipedia.org/wiki/Advanced_Encryption_Standard*
[3] "Advanced Encryption Standard" *Wikipedia* 28 sett 2010 *http://it.wikipedia.org/wiki/Advanced_Encryption_Standard*
[4] "Data Encryption Standard" *Wikipedia* 20 sett 2010 *http://it.wikipedia.org/wiki/Data_Encryption_Standard*

Frontier Foundation collaborarono insieme, per forzare pubblicamente una chiave DES in circa 22 ore e 15 minuti. Inoltre esistono diversi studi teorici, di difficile implementazione pratica, che dimostrano la debolezza dell'algoritmo di cifratura. L'algoritmo acquista maggiore sicurezza pratica nella forma di Triple DES, persistendo tuttavia la sua vulnerabilità teorica. In tempi recenti, il DES è stato superato e sostituito dall'AES (Advanced Encryption Standard).[5]

In alcuni documenti, DES può indicare lo standard di cifratura o indicare l'algoritmo, detto DEA (the Data Encryption Algorithm).[6]

A.2. Cifratura a chiave pubblica

La crittografia a chiave pubblica è un algoritmo di cifratura, la cui caratteristica distintiva è l'uso di algoritmi a chiave asimmetrica in sostituzione o in aggiunta agli algoritmi a chiave simmetrica. Grazie all'uso delle tecniche di crittografia a chiave pubblica, sono diventati disponibili molti metodi pratici per proteggere le comunicazioni o per autenticare i messaggi. Essi non richiedono uno scambio iniziale sicuro di una o più chiavi segrete, come richiesto dagli algoritmi a chive simmetrica. Inoltre questi algoritmi di cifratura possono essere usati per creare firme digitali sicure.[7]

La crittografia a chiave pubblica è una tecnologia che si è diffusa in tutto il mondo ed è alla base di standard di comunicazioni e di autenticazioni usati in Internet, come TLS o Transport Layer Security, il successore di SSL, PGP e GPG.[8]

La tecnica che contraddistingue la crittografia a chiave pubblica è l'uso degli algoritmi a chiave asimmetrica, in cui la chiave usata per cifrare un messaggio non è la stessa per la sua decifrazione. Ogni utente ha una coppia di chiavi — una pubblica ed una privata. La chiave privata è tenuta segreta mentre l'altra è pubblicamente distribuita. I messaggi sono cifrati con la chiave pubblica e possono essere decifrati soltanto con la chiave privata corrispondente. Le chiavi sono matematicamente correllate tra loro ma la chiave privata non può essere facilmente ricavata, in termini di tempo e risorse dalla pubblica. Grazie alla sua invenzione, a partire dalla metà degli anni '70 del secolo scorso, si è sviluppata la crittografia informatica.[9]

In contrasto, gli algoritmi a chiave simmetrica di cui esistono innumerevoli varianti inventate nel corso di centinaia di anni, usano una unica chiave segreta, condivisa, usata sia per cifrare sia per decifrare. In questo schema di cifratura, la chiave segreta deve essere condivisa in anticipo.[10]

Poichè gli algoritmi a chiave simmetrica sono meno avidi di risorse di calcolo, è pratica comune scambiare una chiave usando un algoritmo di scambio chiavi, e cifrare i dati usando questa chiave ed un algoritmo a chiave simmetrica. PGP e la famiglia di protocolli SSL/TLS per esempio, usando questo schema e perciò vengono detti sistemi di cifratura ibridi.[11]

A.2.1. Diffie-Hellman

Lo scambio di chiavi D-H (Diffie–Hellman) è un protocollo di crittografia, che consente a due interlocutori di scambiarsi tra loro una chiave condivisa segreta, su una rete non sicura. Questa chiave può essere usata per cifrare le successive comunicazioni usando un sistema di cifratura simmetrico. [12]

---

[5] "Data Encryption Standard" *Wikipedia.* 20 sett 2010 *http://it.wikipedia.org/wiki/Data_Encryption_Standard*
[6] "Data Encryption Standard." *Wikipedia.* 20 sett 2010 *http://it.wikipedia.org/wiki/Data_Encryption_Standard*
[7] "Cifratura a chiave pubblica." *Wikipedia.* 29 ago 2010 *http://it.wikipedia.org/wiki/Crittografia_asimmetrica*
[8] "Cifratura a chiave pubblica" *Wikipedia.* 29 ago 2010 *http://it.wikipedia.org/wiki/Crittografia_asimmetrica*
[9] "Cifratura a chiave pubblica." *Wikipedia.* 29 ago 2010 *http://it.wikipedia.org/wiki/Crittografia_asimmetrica*
[10] "Cifratura a chiave pubblica." *Wikipedia.* 29 ago 2010 *http://it.wikipedia.org/wiki/Crittografia_asimmetrica*
[11] "Cifratura a chiave pubblica." *Wikipedia.* 29 ago 2010 *http://it.wikipedia.org/wiki/Crittografia_asimmetrica*
[12] "Diffie-Hellman" *Wikipedia.* 17 sett 2010 *http://it.wikipedia.org/wiki/Scambio_di_chiavi_Diffie-Hellman*

A.2.1.1. Storia del protocollo D-H

Lo schema è stato pubblicato la prima volta da Whitfield Diffie e Martin Hellman nel 1976, sebbene si scoprì più tardi fosse già stato inventato alcuni anni prima all'interno del GCHQ (l'agenzia britannica della sicurezza, nonché dello spionaggio e controspionaggio), da parte di Malcolm J. Williamson, ma fino allora tenuto secretato. Nel 2002, Hellmann suggerì di denominare l'algoritmo scambio di chiavi Diffie–Hellman–Merkle, come riconoscimento al contributo apportato da parte di Ralph Merkle, all'invenzione della crittografia a chiave pubblica.[13]

Sebbene lo scambio di chiavi Diffie-Hellman sia un protocollo di scambio anonimo (non-autenticato), esso fa da base per una varietà di protocolli di autenticazione.[14]

Il documento U.S. Patent 4,200,770, descrive l'algoritmo accreditando l'invenzione a Hellman, Diffie, e Merkle..[15]

A.2.2. RSA

In crittografia, l'RSA (RSA sta per Rivest, Shamir e Adleman che per primi lo descrissero pubblicamente), è un algoritmo di crittografia a chiave pubblica. E' il primo algoritmo noto per essere impiegato sia per autenticare sia per cifrare e la sua invenzione ha segnato il primo vero passo in avanti, nel campo della crittografia. L'RSA è ampiamente impiegato nei protocolli di comunicazione digitali, commerciali ed è considerato abbastanza sicuro con l'impiego di chiavi molto lunghe e con implementazioni moderne.[16]

A.2.3. DSA

Il DSA (Digital Signature Algorithm) è uno standard di autenticazione digitale del Governo Federale degli Stati Uniti d'America (o FIPS). E' stato proposto dal NIST (National Institute of Standards and Technology), nell' agosto del 1991 per il suo impiego come standard (Digital Signature Standard o DSS) ed adottato nel 1993, specificato come FIPS 186. Una revisione minore compare nel 1996 specificato come FIPS 186-1. Lo standard è stato ulteriormente esteso nel 2000 come FIPS 186-2 e successivamente nel 2009 come FIPS 186-3.[17]

A.2.4. SSL/TLS

Il TLS (Transport Layer Security) ed il suo predecessore, l'SSL (Secure Socket Layer), sono due protocolli di crittografia che assicurano la sicurezza delle comunicazioni, su reti non fidate come Internet. TLS ed SSL cifrano i segmenti ai capi delle connessioni, al livello del Transport Layer. Diverse versioni del protocollo sono ampiamente impiegate in applicazioni come browser web, client di posta elettronica, fax via Internet, client di chat e applicazioni VoIP (Voice over IP). TLS è un protocollo standard sostenuto dall'IETF, il cui ultimo aggiornamento si trova nel documento RFC 5246, basato sulle precedenti specifiche di SSL, sviluppate da Netscape Corporation.

Il protocollo TLS permette alle applicazioni client/server di comunicare attraverso una rete, impedendo le intercettazioni e le manomissioni da parte di terzi. TLS attraverso la crittografia offre autenticazioni e trasmissioni sicure di dati sensibili tra gli endpoint di una rete non fidata, come Internet. TLS permette cifrature RSA sicure con chiavi da 1024 e 2048 bit.

[13] "Diffie-Hellman." *Wikipedia.* 17 sett 2010 *http://it.wikipedia.org/wiki/Scambio_di_chiavi_Diffie-Hellman*
[14] "Diffie-Hellman." *Wikipedia.* 17 sett 2010 *http://it.wikipedia.org/wiki/Scambio_di_chiavi_Diffie-Hellman*
[15] "Diffie-Hellman." *Wikipedia.* 17 sett 2010 *http://it.wikipedia.org/wiki/Scambio_di_chiavi_Diffie-Hellman*
[16] "RSA" *Wikipedia* 23 ago 2010 *http://it.wikipedia.org/wiki/RSA*
[17] "Digital Signature Algorithm"*Wikipedia* 20 ago 2010 *http://it.wikipedia.org/wiki/Digital_Signature_Algorithm*

In un tipico utilizzo di un browser web, l'autenticazione TLS è unilaterale: soltanto il server è autenticato (il client conosce l'identità del server), il client no (il client rimane non autenticato o anonimo).

Ma TLS supporta anche la più sicura modalità di connessione, bilaterale (tipicamente usata nelle applicazioni enterprise), in cui entrambi gli endpoint della "comunicazione" possono essere sicuri con chi stanno comunicando (a condizione di aver attentamente esaminato le informazioni di identità nel certificato dell'interlocutore). Ciò è nota come mutua autenticazione o 2SSL. La mutua autenticazione richiede che anche il lato client (del TLS) possegga un certificato (con un browser web di solito non si rientra in questo scenario). In alternativa si potrebbero impiegare TLS-PSK, Secure Remote Password (SRP) o altri protocolli in grado di garantire la reciproca autenticazione, in assenza di certificati.

In genere, le informazioni e i certificati necessari per TLS sono gestiti sotto forma di certificati X.509 che impongono requisiti necessari su dati e sul loro formato.

Il protocollo SSL opera in maniera modulare. Per impostazione progettuale, risulta estensibile con compatibilità retroattive e future.[18]

A.2.5. Il sistema Cramer–Shoup

Il sistema Cramer-Shoup è un algoritmo di cifratura a chiave simmetrica che si è dimostrato essere il primo schema efficiente contro attacchi di crittoanalisi basati su assunzioni crittografiche standard. La sua sicurezza deriva dalla intrattabilità computazionale dell'assunzione di Diffie–Hellman. Sviluppato da Ronald Cramer e Victor Shoup nel 1998, esso è una estensione del sistema Elgamal. A differenza di quest'ultimo, estremamente malleabile, Cramer–Shoup aggiunge ulteriori elementi per garantire la non-malleabilità anche contro attacchi molto consistenti. La sua non malleabilità deriva dall'uso di una funzione di hash *collision resistance* e da ulteriore complessità computazionele, risultando in un testo cifrato doppio rispetto a Elgamal. [19]

A.2.6. Cifratura ElGamal

In crittografia, il sitema ElGamal è un algoritmo di cifratura a chiave pubblica basato sul sistema di scambio di chiavi Diffie-Hellman. E' stato descritto la prima volta da Taher Elgamal nel 1985. La cifratura ElGamal viene usata nel software libero GNU Privacy Guard, in recenti versioni di PGP ed in altri sistemi di crittografia. La cifratura DSA è una variante dello schema di autenticazione ElGamal, da non confondersi con la cifratura ElGamal.[20]

[18] "Transport Layer Security" *Wikipedia* 7 ott 2010 *http://it.wikipedia.org/wiki/Transport_Layer_Security*
[19] "Cramer–Shoup cryptosystem" *Wikipedia* 5 October 2010 *http://en.wikipedia.org/wiki/Cramer-Shoup_cryptosystem*
[20] "ElGamal encryption" *Wikipedia* 13 October 2010 *http://en.wikipedia.org/wiki/ElGamal_encryption*

Appendice B. Cronologia Revisioni

Revisione 16.0-1	**Fri September 09 2011**	**Eric Christensen** *sparks@fedoraproject.org*

Branched for Fedora 16.

Revisione 14.3-1	**Sat Apr 02 2011**	**Eric Christensen** *sparks@fedoraproject.org*

Moved VPN text to the Encryption chapter and reformated.

Revisione 14.2-1	**Wed Oct 20 2010**	**Zach Oglesby** *zoglesby@fedoraproject.org*

Added text for using Yubikey on Fedora with local authentication. (BZ 644999)

Revisione 14.2-0	**Fri Oct 6 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Eliminato tutte le variabili nel sorgente del documento.

Revisione 14.1-2	**Fri Oct 1 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Corretto il link a DISA Unix Checklist ed aggiornato link

Revisione 14.1-1	**Wed Jul 8 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto il capitolo su CVE

Revisione 14.0-1	**Fri May 28 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Branched per Fedora 14

Revisione 13.0-7	**Fri May 14 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Rimosso "bug" di testo dal capitolo 7-Zip (bug 591980).

Revisione 13.0-6	**Wed Apr 14 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Completato l'appendice sugli standard di cifratura

Revisione 13.0-5	**Fri Apr 09 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto "Usare GPG in Alpine".

Aggiunto "Usare con Evolution".

Revisione 13.0-4	**Tue Apr 06 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Corretto alcuni problemi riguardanti alcuni paragrafi non traducibili.

Revisione 13.0-3	**Tue Apr 06 2010**	**Eric Christensen** *sparks@fedoraproject.org*

Rimosso il riferimento alla vulnerabilità a PackagKit presente in Fedora 12.

Revisione 13.0-2	**Fri Nov 20 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto la Cronologia Revisioni alla fine del documento.
Aggiunto l'appendice "Standard di Cifratura".

Revisione 13.0-1	**Fri Nov 20 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Fedora 13 branch.

Revisione 1.0-23	**Thu Nov 19 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Ri-aggiornato la sezione "Local users may install trusted packages".

Revisione 1.0-22	**Thu Nov 19 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiornato la sezione "Local users may install trusted packages".

Revisione 1.0-21	**Wed Nov 18 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto la sezione "Local users may install trusted packages".

Revisione 1.0-20	**Sat Nov 14 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto informazioni di Wikipedia all'appendice "Standard di cifratura".
Aggiunto Adam Ligas alla pagina degli autori, per il suo contributo allo sviluppo della sezione "7-Zip".

Revisione 1.0-19	**Mon Oct 26 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiornato la licenza a CC-BY-SA.

Revisione 1.0-18	**Wed Aug 05 2009**	**Eric Chrisetnsen** *sparks@fedoraproject.org*

Risolto il problema relativo al Bug 515043.

Revisione 1.0-17	**Mon Jul 27 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Corretto le informazioni sui rivenditori in SPEC.

Revisione 1.0-16	**Fri Jul 24 2009**	**Fedora Release Engineering** *rel-eng@lists.fedoraproject.org*

Ricompilato per for https://fedoraproject.org/wiki/Fedora_12_Mass_Rebuild

Revisione 1.0-15	**Tue Jul 14 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Aggiunto "desktop-file-utils" a BUILDREQUIRES in spec.

Revisione 1.0-14	**Tue Mar 10 2009**	**Scott Radvan** *sradvan@redhat.com*

Rimosso porzioni di testo più specifiche a rhel e revisioni maggiori.

Revisione 1.0-13	**Mon Mar 2 2009**	**Scott Radvan** *sradvan@redhat.com*

Risolto diversi problemi minori.

Revisione 1.0-12	**Wed Feb 11 2009**	**Scott Radvan** *sradvan@redhat.com*

Nuovi screenshots per F11 in sosituzione di quelli esistenti/datati.

Revisione 1.0-11	**Tue Feb 03 2009**	**Scott Radvan** *sradvan@redhat.com*

Modificato le specifiche LUKS per Fedora 9, incluse quelle delle versioni più recenti.
Risolti alcuni collegamenti a siti web, in particolare i link alla NSA.
Modifiche minori di formattazione.

Revisione 1.0-10	**Wed Jan 27 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Inserito lo screenshot mancante sulla configurazione di un firewall

Revisione 1.0-9	**Wed Jan 27 2009**	**Eric Christensen** *sparks@fedoraproject.org*

Corretti alcuni termini non esatti della fase di validazione. Convertiti in Fedora, precedenti riferimenti Red Hat.